Autonomous Guidance for Asteroid Descent using Successive Convex Optimisation

Dual Quaternion Approach

S. Hazra

Technische Universiteit Delft

TU Delft

DLR

Autonomous Guidance for Asteroid Descent using Successive Convex Optimisation

Dual Quaternion Approach

by

S. Hazra

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday March 28, 2019 at 9:30 AM.

Student number:	4598342	
Project duration:	June 18, 2018 – March 28, 2019	
Thesis committee:	Dr. ir. E. Mooij,	TU Delft, Astrodynamics and Space Missions, Supervisor
	Dr. ir. M. Sagliano,	Deutsches Zentrum für Luft- und Raumfahrt e.V., Supervisor
	Prof. dr. ir. P.N.A.M Visser	TU Delft, Astrodynamics and Space Missions
	Dr. ir. C.D. Visser,	TU Delft, Controls and Simulations

TU Delft

DLR

Abstract

With the onset of the age of space travel, asteroid missions have been steadily gaining interest. The pristine nature of asteroids due to their preserved state since the formation of the Solar System is an opportunity to unravel many mysteries about the Solar System. Also with the ever-growing need for resources, asteroids prove to be a plentiful source. With the discovery of asteroids in the close vicinity of our planet and a probable threat to the preservation of life, a need for defence missions has also arisen. With many successful missions like NEAR to Eros, Rosetta to Comet 67P, Hayabusa to Itokawa, Hayabusa 2 to Ryugu, OSIRIS-REx to Bennu, etc., the requirement of precise navigation with autonomous guidance and control for future missions has been established.

The ever-growing need for better computational speed and accuracy has led to the development of new representations for attitude and position of the spacecraft in the past. The usual methods of representations of position and attitude (pose) are the Cartesian coordinates and quaternions. A recent development is the simultaneous representation of the pose of the spacecraft using *dual quaternions* which are eight-dimensional vectors. As of February 2019, a variety of missions using dual quaternions for relative navigation, rendezvous and docking, entry, descent and landing have been conceptualised. Recent studies have narrowed down from the existing guidance algorithms to, *Successive Convexification* and *Sampling Based Model Predictive Optimisation*, as future prospects for autonomous guidance of missions. These missions are limited by the lack of instantaneous ground control due to operations at very far distances from Earth.

In this thesis, an attempt to incorporate these state-of-the-art guidance technologies to asteroid missions has been made. The novelty in this thesis is, using dual quaternions for SC pose and attitude representation to autonomously guide the spacecraft for mapping an asteroid and perform a touch and go descent using sampling-based motion predictive optimisation and successive convexification respectively. To achieve this objective, this thesis first browses through past missions to establish the requirements of a mission to asteroids. Then it explores the augmented algebra of dual numbers and their application in the concept of dual quaternions. It then works towards establishing the dynamics and kinematics in the asteroid centred rotating frame using dual quaternions along with modelling the environment around the asteroids. This thesis delves into the method of convex optimisation to present its existing algorithms and develop a dynamic successive convex optimisation method to achieve better results as compared to the method developed by Mao et al. (2016). The thesis establishes the initialisation of a sampling-based model predictive optimisation method for autonomous mapping of a target asteroid.

After the development and verification of these algorithms, different scenarios have been designed to find out the robustness of the dynamic successive convexification method. The scenarios have been run by both the Cartesian quaternion and dual quaternion based algorithms. They are found to behave similarly in their results, besides the latter being computationally more expensive as proved by different theses so far. The algorithm performs better than the one developed by Szmuk et al. (2017), but faces difficulties with badly scaled problems as is the nature of missions to asteroids. The software used for the thesis is ECOS, which faces numerical problems in these mission scenarios even when the problem is feasible and has an optimal result. The availability of a solution to the optimal control problem depends on the scaling of the penalty weights used in the cost function to penalise virtual controls and trust region, which are used to prevent infeasibility and bound the problem, respectively. It also depends on finding an appropriate final time for the descent along with the number of nodes for discretisation. This research proves, that successive convexification indeed provides a speedy solution for an autonomous precision descent but needs further work to make it robust and stable. The outcome of this thesis is to carry out further research to understand the complex relations, between the scale of the problem, simulation parameters for the optimal control problem and final time in order to make the algorithm robust and safe for an autonomous mission. Another important future research prospect is to incorporate the sampling based model predictive optimisation, for the SC to autonomously map the target body and help in selecting a landmark for a descent.

Preface

Missions to mere rocky remnants in the Solar System aka asteroids have quickly gained interest in the past few decades as a consequence of the inherent curiosity of human beings. The pristine nature of these celestial bodies has drawn the interest of humans towards landing on them and bringing back samples to find some answers about the Solar System formation process. I am honoured to have had the opportunity to be a part of this research area and contribute the little I could towards the development of such missions.

From sending a probe to one of the gaseous planets to landing on an asteroid, this has been my journey from my very first literature survey to the final one and this report is the culmination of that journey to fulfill the requirements to receive my Master of Science degree from the Aerospace engineering faculty at the Delft University of Technology.

I am immensely grateful to my supervisor Erwin Mooij to have presented me with the opportunity to work under Marco Sagliano at DLR with the thesis topic. With both their guidance, this thesis has taken the present form and I hope someday it will guide a spacecraft to safely descend on an asteroid. Every meeting I have had with Erwin has always been positive and motivating. I am glad he believed in me and had patience with my failures along the journey. At DLR, Marco always found time from his busy schedule to answer my doubts and queries, most of which were pretty basic, but he never showed disappointment. He urged me on, to try a step at a time and kept me on track even when I was back in Delft after my six months in Bremen.

I am thankful for Prof. dr. ir. P.N.A.M. Visser and Dr. ir. C.D. Visser to have agreed to be a part of this thesis as the defense committee. I also thank everyone I have come across and made friends with during this period, especially Bronius who worked on the navigation part of this mission. He always answered my questions as much as he could and provided me with his insights about how navigation instruments work.

I would like to thank all my friends, who kept me sane, grounded and fed. I would be lost without their support and cooking skills. I would like to thank my girlfriend, who has been there for my tears every single time I have lost hope and has been a constant in my world of variables. Lastly, I would like to thank my parents, the two people who were more worried about my nutrition levels than the convergence of my algorithm. Without them, I would not be the person I am today or a graduate to be and for that, I will always be in their debt.

This thesis has tested my capabilities to the extent of making me question my dream of becoming an aerospace engineer, but at the same time has brought me great pleasure in doing the research successfully and I hope you, the reader, will find it useful, interesting and understandable.

S. Hazra
Delft, 2019

Nomenclature

Notation

$\mathbb{C}$	Complex Numbers
$\mathbb{N}$	Natural Numbers
$\mathbb{P}$	Projective Space
$\mathbb{R}$	Real Numbers
$\vee$	Dual number/vector
$\wedge$	Unit vector
$\sim$	Discrete time matrix/vector
$-$	Solution states and controls
$\check{\times}$	Dual vector multiplication
$\otimes, \odot$	Quaternion multiplication
$\check{\otimes}, \check{\odot}$	Dual quaternion multiplication

Latin Symbols

$\boldsymbol{A}$	Jacobian matrix for the states	-
$\mathscr{A}$	Spacecraft space set	-
$\boldsymbol{B}$	Jacobian matrix for the controls	-
$\boldsymbol{C}$	Direction Cosines Matrix	-
$\mathscr{C}$	Configuration space set	-
$\boldsymbol{e}$	Unit direction vector	-
f	Function	-
$f(X)$	Process function	-
$g(X)$	Inequality constraints	-
$h(X)$	Equality constraints	-
H	Half space set	-
$\boldsymbol{I}$	Identity matrix	-
$\boldsymbol{L}_c$	Linear cost function	-
$\hat{\boldsymbol{l}}, \boldsymbol{m}, \theta, d$	Screw axis, Moment, Rotation angle, Distance traversed respectively (Dual Quaternion screw coordinates)	various
$\boldsymbol{n}$	Normal vector	-
$\boldsymbol{N}$	Concatenated virtual control vector	various
$\boldsymbol{N}_c$	Non-linear cost function	-
$\mathscr{O}$	Obstacle space set	-
$\boldsymbol{q}_r$	Rotation quaternion	-
$\boldsymbol{q}_d$	Translation quaternion	-
$\check{\boldsymbol{q}}$	Dual Quaternion	-
$\boldsymbol{Q}$	Stacked Matrices	-
$\boldsymbol{r}$	Position vector	m
$\boldsymbol{r}_e$	Engine thrust vector from COM	m
s, S	Slack Variables	-
$\boldsymbol{u}, \boldsymbol{U}$	Individual and concatenated control vector	-
w	Penalty weight	-
$\mathscr{W}$	World space set	-
$\boldsymbol{x}, \boldsymbol{X}$	Individual and concatenated state vector	various
$\boldsymbol{z}, \boldsymbol{Z}$	Individual and concatenated solution state derivative vector	-
$\boldsymbol{a}$	Acceleration	m/s^2
A	Area	m^2

$\boldsymbol{F}$	Force	N
$\boldsymbol{g}$	Gravitational acceleration vector	m/s^2
G	Gravitational constant	$m^3/kg\,s^2$
$\boldsymbol{J}$	Inertia tensor	$kg\,m^2$
m	Mass	kg
$\boldsymbol{r}$	Position vector	m
t	Time	s
$\boldsymbol{T}$	Torque	Nm
U	Gravitational potential	J/kg
$\boldsymbol{v}$	Velocity vector	m/s

Greek Symbols

α	Exhaust velocity inverse	s/m
$\boldsymbol{\Phi}$	State transition matrix	-
ϱ	Cost ratio (Performance parameter)	-
σ, ς	Scaling factors for penalty weights	-
ϑ	Basis functions	-
$\boldsymbol{\nu}$	Individual virtual control vector	various
$\boldsymbol{\eta}$	Individual trust region radius vector	-
μ	Gravitational parameter	m^3/s^2
ρ	Density	kg/m^3
γ	Glide-Slope angle	rad
ι	Line of sight angle	rad
δ	Spacecraft tilt angle	rad
θ	Engine gimbal angle	rad
$\boldsymbol{\omega}$	Angular velocity vector	rad/s
ω_f	Solid angle	rad
$\boldsymbol{\Omega}$	Angular velocity tensor	-
$\boldsymbol{\Xi}$	Quaternion tensor	-

Subscripts

$\mathtt{A}, \mathtt{B}, \mathtt{I}$	Asteroid, Body and Inertial frame respectively
$\boldsymbol{x}^{\mathtt{B}}_{\mathtt{A/I}}$	Value of $\boldsymbol{x}$ for A frame with respect to I frame in the B frame
$\boldsymbol{x}^{i}_{k}$	Value of $\boldsymbol{x}$ at time step k and at iteration i

Superscripts

$^*, ^\bullet, ^\diamond$	Conjugates
$^{-1}$	Inverse
$^{\mathtt{T}}$	Transpose

Abbreviations

AOCS	Attitude and Orbit Control System
ADCS	Attitude Determination and Control System
AU	Astronomical Unit
COM	Centre of Mass
CQ	Cartesian-Quaternion
DCM	Direction Cosine Matrix
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
DOF	Degree of Freedom
DQ	Dual Quaternion
DQEKF	Dual Quaternion Extended Kalman Filter
DQMEKF	Dual Quaternion Multiplicative Extended Kalman Filter
ECOS	Embedded Conic Optimization Solver

EKF	Extended Kalman Filter
EOM	Equations of Motion
ESA	European Space Agency
FDIR	Fault Idnetification, Detection and Recovery
FOV	Field of View
GCP	Global Control Point
GG	Gravity Gradient
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
HDA	Hazard Detection and Avoidance
HGA	High Gain Antenna
IMU	Inertial Measurement Unit
IRU	Inertial Reference Units
IVP	Initial Value Problem
JAXA	Japan Aerospace Exploration Agency
JPL	Jet Propulsion Laboratory
LHS	Left Hand Side
LIDAR	Light Imaging, Detection, And Ranging
LOS	Line of Sight
LR(F)	Laser Ranger (Finder)
LTV	Linear Time Variant
LVLH	Local Vertical Local Horizontal
MBA	Main Belt Asteroids
MC	Monte Carlo
MEKF	Multiplicative Extended Kalman Filter
MEMS	Micro-Electro-Mechanical System
MVM	Mission Vehicle Management
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
NAVCAM	Navigation Camera
NEA	Near-Earth Asteroid
NEAR	Near Earth Asteroid Rendezvous
OCP	Optimal Control Problems
OD	Orbit Determination
ONC	Optical Navigation Camera
QVEKF	Quaternion-Vector Extended Kalman Filter
RADAR	RAdio Detection And Ranging
RHS	Right Hand Side
RK45	Runge-Kutta 45
SBMPO	Sampling Based Motion Planning Optimisation
SC	Spacecraft
SCvx	Successive Convexification
SE(3)	Special Euclidean Group in 3D
SO(3)	Special Orthogonal group in 3D
SOCP	Second Order Cone Programming
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events
SRP	Solar-Radiation Pressure
STT	Star Tracker
TAG	Touch and Go
TNO	Trans-Neptunian Objects
TPBVP	Two-Point Boundary Value Problem
TRM	Trust Region Method
YORP	Yarkovsky–O'Keefe–Radzievskii–Paddack effect

Contents

1

Introduction

Asteroids are the remnant debris from the planetary formation phases during the initialisation of the Solar System around 4.6 billion years ago. The debris left from the phase of the runaway growth were perturbed by the newly formed planetary embryos that were cleaning up the gas and matter in their path. The gravity of newly formed Jupiter brought an end to the formation of planetary bodies in the region near it and caused the small bodies to collide with one another, fragmenting them into the asteroids we observe today. They can, therefore, provide details of the process of core formation or the composition of the protoplanetary disk due to the different condensates present at varying distances from the protosun. Furthermore, collisions of such asteroids on Earth could be its source of water. They could also provide information to establish possible mechanisms for the formation of the inner planets. Figure 1.1 provides a visual of the presently believed procedure of the formation of the Solar System.

The past decade has witnessed quite a few flybys of asteroids and missions to them for close proximity operations. The Hayabusa mission was hindered by an inadequately developed autonomous capability (Kubota et al. (2006)) to collect enough samples from the surface of Itokawa. It also faced unexpected interactions between the autonomous and ground station commands resulting in an inaccurate deployment of the science package MINERVA, that escaped the asteroid altogether and was lost in space. The industry also witnessed the loss of Philae, Rosetta's lander for Comet 67P as its energetic rebound on landing on the surface resulted in its hard landing in an unfavourable location. This cost the mission its science objectives to be fulfilled

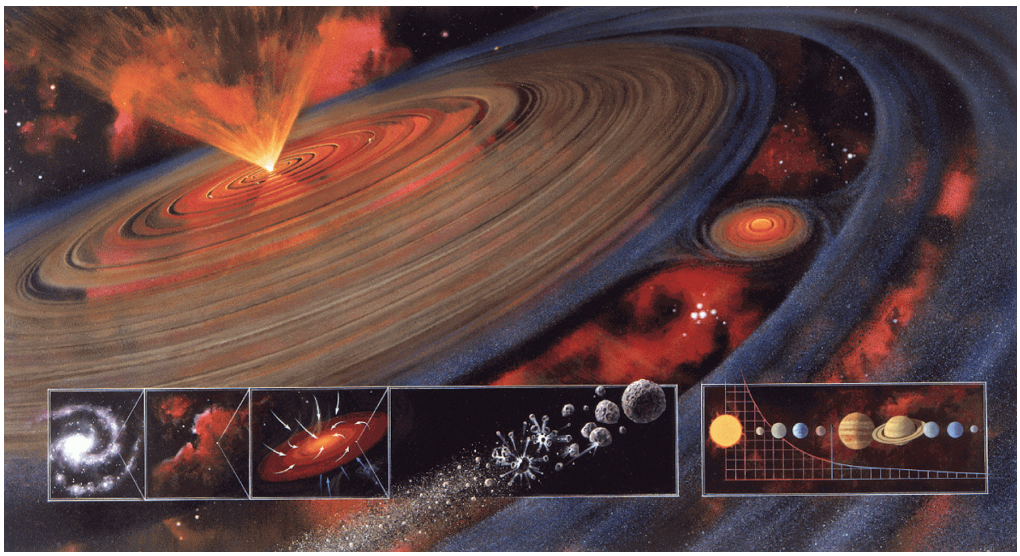


Figure 1.1: An artist's rendition of the planetary formation process. [1]

[1] http://www.agnld.uni-potsdam.de/ frank/research.html; date accessed: 04-10-18

at the surface. These failures have already paved the way towards the need for advanced guidance, navigation and control (GNC) capabilities to be able to handle the challenges posed by the complex and perturbed environment of the small Solar System bodies.

Missions to asteroids, deal with quite a few challenges: their distance from the Earth, their variable or fast rotation rates, irregular shapes and non-uniform densities, the effect of the disturbing forces of planets and the Sun on them, etc. The development of affordable high-end technology for a mission to the small Solar System bodies is, therefore, a requirement of the present mission designs.

1.1. Asteroid Mission Objectives

Asteroids, as we mentioned above, are therefore of fundamental scientific importance. Missions to asteroids serve the following high-level objectives:

- **Science:**
  Being the remnants of the Solar System formation and having their conditions preserved, from their different compositions, we can find out the different conditions during the formation, respective to their distance from the Sun.[2]

- **Future Resource:**
  Asteroids are the closest substantial resource beyond the Earth for scarce materials. Asteroid mining is, therefore, a future objective to quench our needs. These objects are classified based on their spectral type. The C-type, dark and carbon-rich, have phosphorous and other elements required in fertilisers, high abundances of water that could provide for in-situ utilisation as rocket propellant. S-types contain significant fractions of iron, nickel, and cobalt. They also have a good ratio of rare metals like platinum and gold. M-class asteroids have ten times more metal than the S-types, but pose a high cost of the mission due to their distance and rarity.[3]

- **Planetary Defence:**
  In the past, Earth has been a subject to asteroid impacts. The Lunar and Planetary Laboratory at the University of Arizona provides the fact that Earth has more than three million impact craters larger than 1 km in diameter, the largest being 160 km in diameter in South Africa.[4] In 2013, an asteroid of mere 20 m in diameter sent a shockwave striking six cities across Russia, the Chelyabinsk airburst.[5] Although rare, larger impacts could lead to major natural catastrophes. NASA and ESA both have been considering the use of space missions for asteroid risk assessments.

1.2. Autonomous Guidance

The Main Asteroid Belt (discussed in detail in the Chapter 2) lies between 2.1 to 3 au, between Mars and Jupiter and holds small Solar System bodies of different compositions. Therefore, this is the perfect site to find a scientifically interesting small Solar System body for close proximity operations. With some rough calculations with their distance from the Earth, we realise that back and forth communications from the ground station (GS) to the target body would take from $\approx$ 17 to 33 mins. Commands from the GS can be used for remotely operate the spacecraft (SC), but with the time delay involved, this could prove to be highly costly. Also in case of unforeseen circumstances, the SC is impaired without autonomy and the time delay for active control could jeopardise the safety of the SC.

The gravity fields of the small Solar System bodies are highly perturbed and non-central due to their irregular shape or differentiated composition and very weak. Due to this stable near-circular orbits could turn into crashing or escape orbits in days and this poses a problem. They are greatly subject to perturbations by nearby object approaches and solar radiation pressure (SRP). Also, some of them are subject to fast rotation periods. As a target for a mission, they have the most perturbed environment and hence the dynamics around them can be unpredictable. For long duration close proximity operations, despite being weak, the perturbing accelerations increase the fuel cost for a simplified trajectory design.

Additionally, there are restrictions on minimum volume and mass of the SC, which makes the designing, constructing and testing of such an intricate system an expensive one (Ley et al. (2011)). The SC, for any kind

[2] https://solarsystem.nasa.gov/planets/asteroids/indepth; date accessed: 01-10-18

[3] https://www.nasa.gov/content/goddard/new-nasa-mission-to-help-us-learn-how-to-mine-asteroids; date accessed: 01-10-18

[4] https://en.wikipedia.org/wiki/List_of_impact_craters_on_Earth, date accessed: 28-02-2019

[5] http://m.esa.int/Our_Activities/Space_Engineering_Technology/Asteroid_Impact_Mission/Planetary_defence; date accessed: 01-10-18

of mission, has finite limits on the electrical energy, thrust or propellants. These limits drive the requirements of efficient and accurate equipment for functions like steering and navigating along the planned path. Also for a mission to asteroids, mapping and analysing them while staying in orbit around them for landing at the right location to extract material for both scientific and mining purposes.[6]

All these challenges can be handled well with improved autonomy of the SC and one of the key technologies to help achieve this is, autonomous guidance, a robust and optimal one. The function of the guidance system is to calculate the desired attitude or trajectory of the SC by optimising certain parameters like propellant consumption, time of flight, etc. It is, therefore, nothing, but a mathematical black box, that contains algorithms to compute the desired attitude and rate based on the current mode, time and function of the SC. This desired output is then compared to the actual attitude and rate of the SC provided by the navigation system, and the error is computed. The guidance system then generates the required command to correct the attitude, which is then received by the control system of the SC.

Some very promising recent works include the ones by Szmuk et al. (2017) and Szmuk and Açıkmeşe (2018) that have incorporated successive convex optimisation to handle non-convexities and non-linearities in the Mars descent scenario and Capolupo et al. (2017), Surovik and Scheeres (2017) that use motion planning algorithms for mapping small Solar System bodies while dealing with complicated observational performance requirements. These methods have proved to be promising and are therefore motivation for research. They have been explored in depth in Chapter 6 and Appendix A and have been incorporated for the development of this thesis.

1.3. Dual Quaternions

For GNC the position and the attitude of the SC are represented in different forms: Cartesian, Keplerian or non-singular parameters. Quaternions and Modified Rodrigues Parameters (MRPs), which are non-singular parameters (MRPs for all angles besides 2π), have proved to be the best way to achieve an efficient, robust and reliable system so far. Quaternions are popular due to their compactness, saving onboard memory and also being singularity free, enabling the transformation of attitude by rotation about the Euler axis.

Dual quaternions are a step ahead in the representation of pose (position and attitude) that combines the concept of dual numbers and quaternions. A dual quaternion contains two quaternions, one that represents rotation and the other translation. As can be guessed, the kinematics or dynamics formed with dual quaternions would represent both the translation and rotation of the SC. The easiest way to visualise a dual quaternion is to discern the motion of a screw. When the SC is operated by a dual quaternion, it could be imagined that the SC was being translated and rotated about a particular axis just like the motion of the screw. The mathematics governing dual quaternion operations is discussed in Chapter 3.

Although being complex to understand, their superior compactness compared to quaternions with the kinematic and dynamic equations has increased the interest in them. They also have properties similar to regular quaternions, which makes their operations more lucrative. It was, however, yet to be proven at the beginning of this research to be computationally more efficient than quaternions by equivalent or analogous comparison and needed further research work to be included as a beneficial non-singular parameter. It was hence incorporated in this thesis as the attitude and pose representation parameter.

The most exciting researches done recently with dual quaternions include the work by Razgus et al. (2017) who have used dual quaternions for relative navigation, Ballester (2018) who has studied the advantages and disadvantages of using dual quaternions for pose and attitude representation, Lee and Mesbahi (2015), and Lee and Mesbahi (2017) who have implemented model predictive control using dual quaternions for an autonomous precision landing problem. Utilising these works as a baseline, we have approached the guidance problem for asteroid descent using dual quaternions.

1.4. Research Questions

The discussions from the sections above lead to the three regions of interest, that combine towards a question to be researched for this thesis. Asteroids being the future destination for space missions, need to be studied in situ as well as brought back as samples for further research. The guidance system for an asteroid mission needs to be developed to run autonomously and enable the SC to meet its scientific and mission objectives on time in real time. For autonomous guidance, computations need to be faster; with smaller onboard memory

[6] http://m.esa.int/Our_Activities/Space_Engineering_Technology/Asteroid_Impact_Mission/Planetary_defence; date accessed: 01-10-18

utilisation and this calls for better and better ways of pose representation. This is where dual quaternions come in the picture. All these lead to the following primary question of interest as motivation for research:

How can a spacecraft autonomously descend for a touch-and-go sampling process to a chosen landmark robustly and optimally, while being accurate and safe?

To answer this question we will address and research the following sub-questions:

- How can the SC attitude and orbital dynamics be represented by dual quaternions?
- What are the cost functions that need to be optimised, the constraints they are subject to and how can they be formulated with dual quaternions?
- How can the method of successive convex optimisation (SCvx) be implemented to optimise the derived cost functions?
- What methods can be used to verify and validate the subsystems and guidance algorithm to qualify the system to be accurate, safe and robust?
- How useful is this representation and can it be used in real time autonomous applications?
- How can the method of sampling based model predictive optimisation (SBMPO) be used to map a small Solar System body autonomously to provide the SC with a safe landmark for descent?

To the best of the author's knowledge, this thesis is the first to incorporate a highly constrained SCvx guidance algorithm with DQs and a realistic gravity model of asteroids to better understand the advantages and disadvantages of a conjunction of these methods for its application as a state-of-the-art autonomous guidance technology.

1.5. Report Outline

The report consists of ten chapters, including the introduction. The organisation is chosen such that the required concepts are discussed sequentially to help the reader understand the concepts before approaching this research for future work. A brief overview of each chapter is discussed below to help the reader understand the gist of the report.

- **Chapter 2**
  In this chapter, we talk about asteroids, some relevant missions to them and some recent research using dual quaternions and the target mission for this thesis. We also discuss and establish the top level initial mission objectives, requirements, and assumptions.
- **Chapter 3**
  In this chapter, we discuss the concept of dual quaternions. We discuss the mathematical concept of dual numbers, their arithmetic and how they have been incorporated to represent both position and attitude of the SC. We also discuss the operations that have been established with dual quaternions and provide an example for insight as to how they work.
- **Chapter 4**
  This chapter is dedicated towards everything that is required for attitude and orbital dynamics of the SC and the asteroid. It discusses attitude representations, frame conversions, converting DQ inertial kinematics and dynamics to the relative frame and also the external disturbance mathematical models.
- **Chapter 5**
  The fundamental theory for autonomous guidance is discussed here. We also establish the high-level optimisation problem for mapping and descent guidance in this chapter by discussing the requirements for the mission in detail.
- **Chapter 6**
  In this chapter, we establish all the required concepts for convex optimisation and the final problem formulation for the descent optimal control problem using successive convex optimisation and DQs.

- **Chapter 7**
  We present the simulator in this chapter with all the required system and subsystem verification.
- **Chapter 8**
  This chapter holds all the results from the simulator for different target mission scenarios.
- **Chapter 9**
  This chapter concludes the thesis with takeaways from this research and recommendations for future work.

The basic concepts of sampling based motion planning, sampling based motion predictive optimisation, its initial implementation and simulations have been presented in Appendix A. Due to time constraints the full implementation of this method could not be worked out.

2

Asteroids and Mission Heritage

The first asteroid flyby was in 1991 by the spacecraft Galileo across the asteroids Gaspra and Ida on its way to Jupiter. Since then there have been a number of missions, that have been dedicated to not just flybys, but landing on and studying asteroids. There is an increased interest in them in recent years due to the objectives discussed in the previous chapter. In this chapter, we will briefly discuss the small Solar System bodies categorised as asteroids and some of the most recent asteroid missions. A brief discussion on relevant research work done on the combined topics of asteroid missions, dual quaternions and optimised guidance and control has also been presented in this chapter. Lastly, we establish the mission and system requirements and assumptions required for this thesis.

2.1. Asteroids

According to the new classification by the International Astronomical Union (IAU), celestial bodies are classified as planets, dwarf planets, and small Solar System bodies. Any object that has a size smaller than Mercury and has enough gravity to hold a spherical shape is classified as a dwarf planet.[1] Subsequently any object not massive enough for its gravity to hold a spherical shape is classified as a small Solar System body. Most of the Solar System asteroids fall in this group. Almost one and a half million asteroids are now known with an average separation of greater than 1-3 million km between each other.[2] Figure 2.1 provides an image of small Solar System bodies, that have had different spacecraft flyby them or landed on them.

2.1.1. Classification by Location

Asteroids have elliptical orbits around the Sun and rotate about themselves. Quite a number of asteroids have been found to have smaller asteroids or moonlets revolving around them. Binary and tertiary systems have also been observed, where two or three similar sized asteroids orbit around each other as a system.[3] They are classified depending on their location in the Solar System as given below.

Near Earth Objects (NEO)
Asteroids that have orbits with perihelia 1.3 au causing a close approach with the Earth fall in this category. These attract much attention due to the danger they pose due to the possibility of collisions with Earth. There are four subcategories: *Amor, Apollo, Atens and Apohele* based on there perihelia(p), aphelia(q) and semi-major axes(a).

- **Amor:** (1.017 au < p < 1.3 au) They are outside the Earth's orbit, but within Mars's.
- **Apollo:** (p < 1.017 au and a > 1 au) They spend most of their time away from the Sun than Earth.
- **Atens:** (a < 1 au and q > 0.983 au) They are the Earth-crossing asteroids and spend most of their time closer to the orbit of the Earth.
- **Apohele:** Orbits completely within the Earth's orbit. These asteroids are difficult to detect

[1] https://www.iau.org/public/themes/pluto; date accessed: 30-09-17
[2] https://solarsystem.nasa.gov/planets/asteroids/indepth; date accessed: 30-09-17
[3] Same as footnote 2

NEOs have a short dynamical lifetime of $\leq 10^7$ years and get ejected into space or are tidally or thermally destroyed by the Sun. They need to be replenished and their source is the main asteroid belt.

Main Belt Asteroids (MBA)
The main belt asteroids lie between 2.1 - 3.3 au from the Sun, between Mars and Jupiter. Most asteroids are in this belt. The orbits of these asteroids are a subject to the strong gravitational influence of Jupiter. The giant planet causes perturbations leading to chaotic zones around resonance locations. This can cause the eccentricity of the orbits of the inner belt of asteroids to increase enough to have close approaches to Mars and Earth or even the Sun. They can thereby be removed from orbit by the gravitational interactions or made to collide with terrestrial planets.

Trans-Neptunian Objects (TNO)
As the name suggests these asteroids lie beyond the orbit of Neptune and account for the majority of small bodies in the Solar System. This is also known as the Kuiper belt. According to the IAU's new classification, Pluto and its moon Charon are a part of this belt. They contain the classical Kuiper belt objects, that have low eccentricities and the scattered disk objects with high eccentricities.

Trojans
Asteroids that lie in the same orbit as the planet near the Lagrangian points L_4 and L_5 of the planet and the Sun are known as Trojan asteroids. These points are approximately 60° ahead and behind the planet known as Greek camp and Trojan camp respectively. Jupiter has the most number of trojans, believed to be almost a million in the count. 6000 of them have been already identified. A few Neptune (17) and Mars (4) trojans have also been found (Sheppard et al. (2006)). Saturn and Uranus are also believed to have trojans based on numerical calculations of the involved orbital dynamics. Earth's first trojan, 2010 TK7 was found in 2011.[5]

Centaurs
Centaur asteroids orbit between the orbits of Jupiter and Neptune and sometimes also have chaotic planet-crossing orbits. Their dynamical lifetime is around $10^6 - 10^8$ years. They are believed to be transitioning from mostly the TNO's scattered disk objects.

Vulcanoids
It is believed that there are asteroids between 0.07 to 0.21 au from the Sun. To date, not a single Vulcanoid

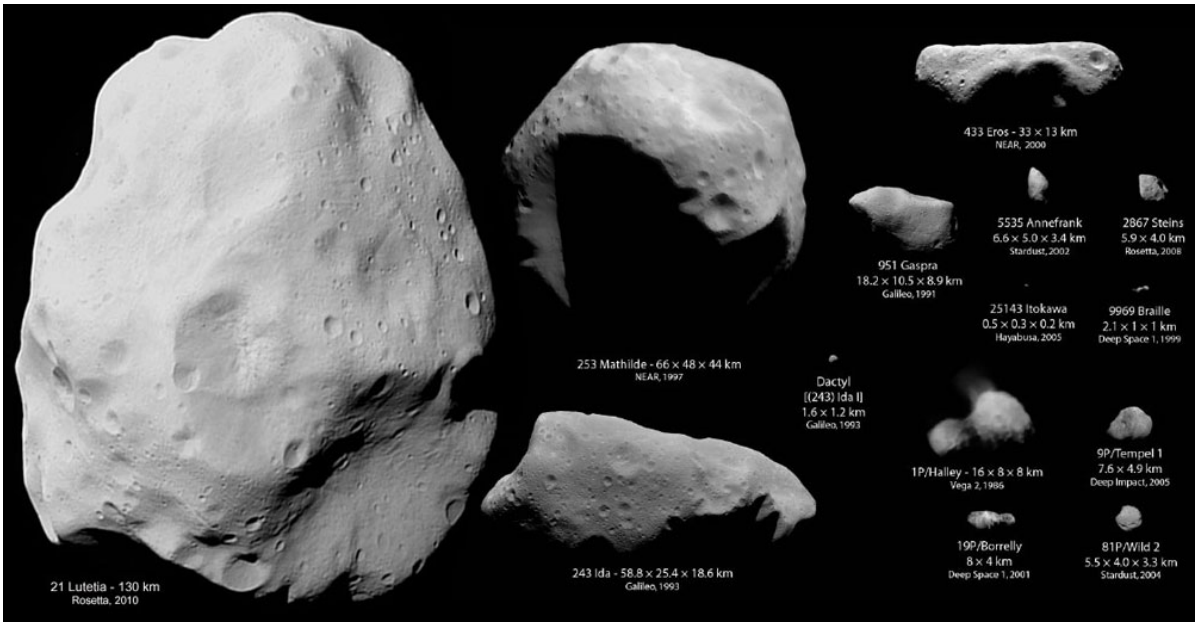


Figure 2.1: Images of small Solar System bodies from flyby missions.[4]

[4]http://www.russianspaceweb.com/spacecraft_small_bodies.html; date accessed: 04-10-17
[5]http://www.nature.com/nphys/journal/v7/n8/full/nphys2061.html?foxtrotcallback=true; date accessed: 01-10-17

has been observed. Even if they exist, being so close to the Sun they would be difficult to find. So they remain hypothetical till proven right by observation.

2.1.2. Classification by Composition

The composition of asteroids as discussed in the previous chapter depends on their distance from the Sun. The variation in the undergone thermal processing has led to the existence of a compositionally diverse group of objects. For example, ones that faced higher temperatures have iron centres, since it melted and sunk in forcing basaltic lava to surface. Vesta is one such asteroid. They are classified based on their composition as carefully inspected from their spectral images accounting in the space weathering.[6] The three main types with respect to composition are the C, S and M type.

Carbonaceou/SC-type
The C-type is the largest class of asteroids; roughly 40% of the known asteroids. These asteroids are dark and have flat spectra. They seem to be made of low-temperature condensates, that have undergone little or no heating. The appearance indicates the presence of clay and silicate rocks.

Stony/S-type
The next largest class of asteroids; around 30-35%, is this one. They are fairly bright and appear to consist of silicates and iron-nickel.

Metallic/M-type
M-type asteroids have spectra that show a composition of nickel-iron embedded in enstatite, which is a magnesium-rich silicate. These asteroids have undergone a melt phase due to substantial thermal processing and could also possibly be cores of disrupted larger asteroids.

D & P-types
D & P-type asteroids do not exhibit any spectral features and could be the most primitive class of asteroids. They appear to be redder than the S-type asteroids and this has been attributed to the possible formation of organic compounds due to space weathering.

Water/W-type
W-type was a reclassification from M-type when many of the larger asteroids(≥30 km) showed an absorption band implying the presence of water in the form of hydration.

Vesta/V-type
As said earlier, the asteroid Vesta is one of a kind due to its size, to have a surface covered by basaltic material. Smaller objects in similar orbits to Vesta have been found to have a similar composition. They could have possibly been eroded off from Vesta due to an impact collision.

2.1.3. Properties of Asteroids

The shape of a solid minor body depends on its mass, composition, and temperature. It also depends on its spin rate and tides. Depending on the internal strength of the material and its density, the body has to have a certain minimum radius to be spherical. If the body is composed of silicates then the minimum radius has to be approximately 350 km, whereas for a metallic body the minimum radius is merely 100 km (Lissauer and Pater (2013)). Except for the dwarf planets, asteroids are very irregular in shape. These shapes also depend on collisions and accretion of material leading to abnormal shapes and differentiated compositions. (Holsapple (2001)) have proved that such cohesionless bodies can exist in equilibrium with a number of possible shapes at a particular spin rate. The asteroids also undergo changes in their shape due to weathering. The Yarkovsky–O'Keefe–Radzievskii–Paddack effect (YORP) changes the rotation rate of small bodies and are responsible for creating tumbling and binary asteroids hence modifying their heliocentric radial drifts.[7] As these bodies gain momentum from the YORP torque, their shapes change. Simulations were made for this torque, till landslides occur on the body and the stable shape acquired is like a 'top' resembling several such critically tumbling bodies imaged by RADAR (Harris et al. (2009)).

They range from big sizes like Ceres, that is around 587 km in diameter to ones that are smaller than 10 m in diameter.[8] Being less massive they have weak gravity fields. Hence they cannot sustain and do not have

[6] It is the interaction of solar wind particles, solar radiation and cosmic rays with asteroids
[7] https://en.wikipedia.org/wiki/Yarkovsky–O'Keefe–Radzievskii–Paddack_effect; date accessed: 01-10-17
[8] https://solarsystem.nasa.gov/planets/asteroids/indepth; date accessed: 01-10-17

atmospheres. Their surface is usually pitted or cratered. The irregularity in shape causes erratic rotation and their gravity field to be perturbed. Due to this a central-gravity field model cannot be used for asteroids. Due to their weak gravitational field forces like gravitational attraction by other celestial bodies, solar radiation and electromagnetic force can perturb them from their Keplerian orbits.

Asteroids have a varied range of spin rates. Mostly they gained angular momentum from collisions (Harris et al. (2009)). Depending on their size, shape, and composition they have an upper limit to their spin rate. If above this limit, the centrifugal force would exceed its self-gravitational force scattering itself in space. It is seen that no small body larger than 100 km in diameter has a rotation rate of fewer than 2 hours. But we do observe smaller bodies rotating with periods as small as 0.0222 hours.[9] This says that the bigger objects are an aggregate of rocks or differentiated than solid, whereas the smaller bodies being monolith have more tensile strength rather than gravity keeping them together. Spin rates of asteroids are affected by close encounters with planets, the Yarkovsky effect as well as its variant the YORP effect.

2.2. Mission Heritage

Missions to small Solar System bodies deal with an uncertain a priori knowledge of the target object; its shape, size, rotation rate, kinematics, orbit dynamics and other factors. This led to the development of autonomous navigation and advanced guidance and control techniques for accurate pointing and landing; serving the mission's science objectives. This section is dedicated to the discussion of some state-of-the-art missions till September 2017, that observed and landed on small Solar System bodies. It also discusses certain research work directed towards the utilisation of dual quaternions for the development of autonomous guidance and control of spacecrafts for observing and landing on celestial bodies relevant to asteroid missions.

2.2.1. Asteroid Missions

A few of the most recent asteroid missions are JAXA's Hayabusa and Hayabusa 2, ESA's Rosetta, NASA's DIXI and NASA's OSIRIS-REx. Hayabusa 2 and OSIRIS-REx are still ongoing missions, but the onboard GNC strategies used are relevant for the thesis research.

Hayabusa/Hayabusa 2

This was the first sample return mission from an asteroid undertaken by JAXA, launched in May 2003 to the asteroid Itokawa. Hayabusa collected some material from the asteroid surface on its first touchdown, but the second attempt faced operational problems. It returned to Earth with the samples in June 2010. Hayabusa 2 is an extension of Hayabusa and was launched in December 2014 to the asteroid Ryugu (1999 JU3). It is expected to arrive at the target in July 2018. It is planned to survey the asteroid for a year and a half and then return back with samples in December 2020. Three sampling touchdowns and a 2 m crater generation by high-speed impact operation have been planned for the proximity operation.

Hayabusa used a multi-band imaging camera to image the entire surface of the asteroid whilst its tracking and navigation data provided with a mass and volume estimate of the asteroid (Fujiwara et al. (2006)). It had an autonomous onboard guidance and navigation system to touch down on Itokawa, since the exact shape, size and surface conditions of the asteroid were unknown. The onboard GNC system consisted of a two axes Sun sensor, star tracker and an IRU for attitude determination, an accelerometer and reaction control system with thrusters and reaction wheels for attitude and position control. Figure 2.2 provides the GNC functional block diagram. Navigation was carried out by means of a narrow-angle and wide-angle cameras. for mapping and for regional safety monitoring of surface obstacles. High altitude and low altitude measurements were done by a lidar and a laser ranger respectively. The AOCS used the inputs from the cameras, lidar, laser ranger and an extended Kalman filter for state estimation. The design of the GNC system of Hayabusa 2 is significantly similar to Hayabusa. The instruments for navigation are just upgraded versions as those of Hayabusa. The AOCS consists of two redundantly equipped attitude and orbit control processor units, two-star trackers, two IRUs, four accelerometers, four coarse Sun sensors and four reaction wheels (Tsuda et al. (2013)).

Rosetta

ESA's Rosetta was launched in March 2004 to the Comet 67P/Churyumov-Gerasimenko. It performed flybys of asteroid Steins and Lutetia before reaching the comet in August 2014. It mapped the comet from an

[9] https://web.archive.org/web/20060512060350/http://spaceguard.iasf-roma.inaf.it/tumblingstone/issueSCurrent/eng/ast-day.htm; date accessed: 01-10-17

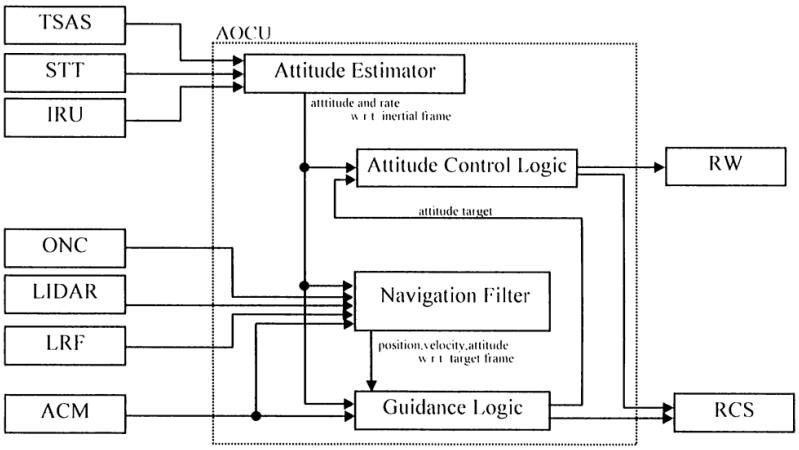


Figure 2.2: Functional Block Diagram of Hayabusa GNC function (Kubota et al. (2006)).

orbit around it before sending the lander, Philae to its surface. After being in space for more than 12 years Rosetta's mission concluded with it descending on the surface of the comet in September 2016.

Rosetta's GNC system consisted of two pairs of coarse Sun sensors, two star trackers, three IMUs with three gyros and accelerometers each and two navigation cameras (NAVCAM) for navigation and for control two sets of twelve thrusters of 10 N each, a bi-propellant system, four-momentum wheels, 1-DOF solar array pointing mechanisms and a 2-DOF antenna pointing mechanism. Similar to Hayabusa, a priori knowledge of the comet kinematics and dynamics was uncertain and hence needed to be obtained in situ during navigation. Radiometric tracking from ground stations of the SC's trajectory using range and Doppler measurements prove insufficient due to this uncertainty and therefore a relative state needed to be measured to improve relative navigation performance. OSIRIS-NAC and NAVCAMS provided optical images to detect the asteroid of interest and deduce the relative trajectory from Rosetta to the centre of the asteroid of interest or to a recognisable feature on its surface. Three separate computer programs were used for orbit determination (OD): first; SC OD using radiometric data only, second; object OD using astrometric data only and third; SC-object relative state estimation using optical measurements of the direction from SC to body centre taking the first two OD solutions (Muñoz et al. (2012)). For the comet, there were four observational phases to achieve navigation accuracy for the lander delivery phase. Figure 2.3 provides a montage of images by Rosetta during the descent and topple of Philae.

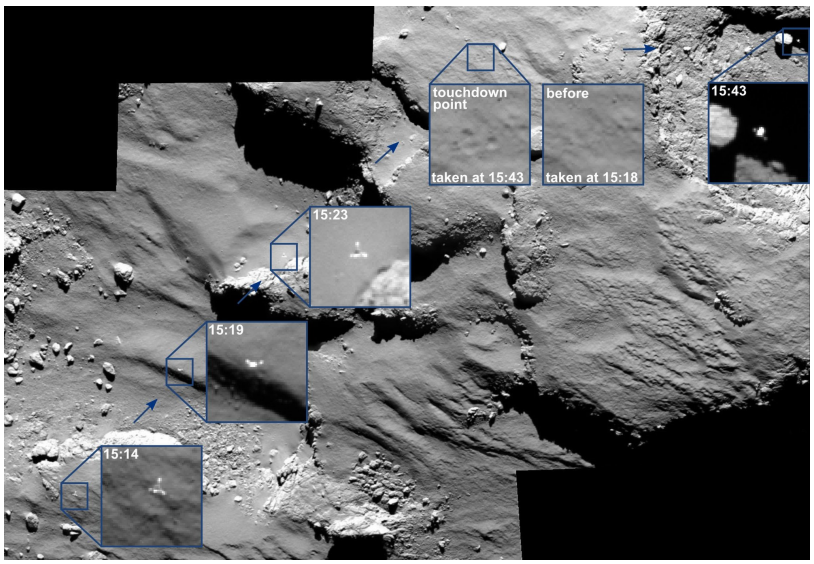


Figure 2.3: Rosetta tracking Philae's descent and topple on Churyumov-Gerasimenko. Image Courtesy: ESA

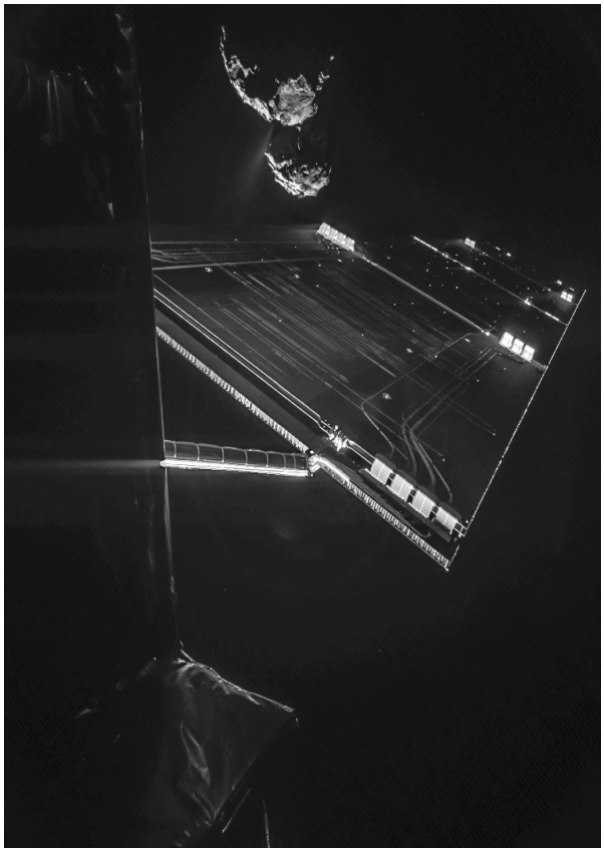


Figure 2.4: Rosetta taking a selfie with Comet Churyumov-Gerasimenko. Image Courtesy: ESA

Deep Impact Extended Investigation (DIXI)
NASA designed a probe to study the composition of the Comet Tempel 1 and launched it in January 2005. DIXI successfully launched an impactor at the nucleus of the comet, ejecting a debris cloud for the first time from a comet's surface. Figure 2.5 elaborates the flyby and impactor trajectories on the encounter.

The flyby SC was equipped with two instruments the High-Resolution Instrument (HRI) combined with a filtered CCD camera and IR spectrometer and the Medium Resolution Instrument (MRI) and the impactor was equipped with Impactor Targeting Sensor (ITS), which is a duplicate of the MRI camera (Hampton et al. (2005)). The nucleus was imaged by the flyby SC for days before the impact, but its shape was incompletely determined due to its slow rotation period and high speed of the flyby. Observations from the impactor saw numerous outbursts from the surface as expected in the close vicinity of the Sun. The impact was near the southern limb of the nucleus and was an oblique one. The flyby SC was made to slow down by 100 m/s and oriented such that it could look back to image the impact. These look-back images helped observe and determine the lateral expansion of the ejecta cone and its ballistic path back to the surface and the local gravitational field of the nucleus respectively.

MRI was used for autonomous navigation during the encounter and HRI in the approach phase by the flyby SC. An S-band antenna was used for communication with the impactor. A three-axis stabilised momentum wheel-based control system along with four RCS thruster hydrazine propulsion for TCMs and momentum dumps was used for its control. The ADCS subsystem estimated and controlled the SC's attitude, rates and linear accelerations with the help of two-star trackers and an IRU and provided the information to the AutoNav system to determine the pointing of the navigation camera's boresight (Mastrodemos et al. (2005)). The impactor itself had a three-axis stabilised rate control system (RCS) with four RCS thrusters and an ADCS system with a star tracker and an IRU along with ITS. The AutoNav used a combination of images from the impactor and its attitude telemetry to determine its impact site. A few problems were faced by the impactor when large dust particles hit it and slewed away it is pointing, but the ADCS subsystem brought it back centering on the impact site.

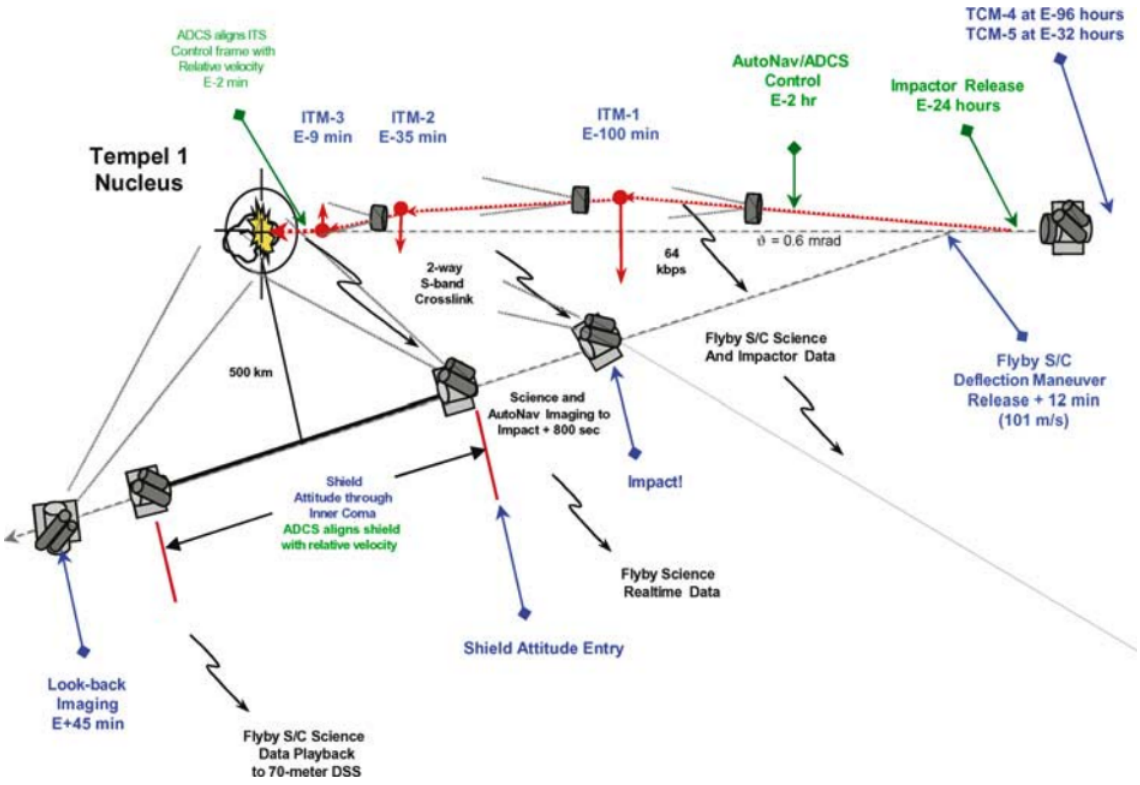


Figure 2.5: Tempel 1 encounter schematic for Deep Impact mission (Mastrodemos et al. (2005)).

OSIRIS-REx

NASA launched the Origins Spectral Interpretation Resource Identification Security - Regolith Explorer (OSIRIS-REx) in September 2016 to the asteroid Bennu. It is presently on a seven-year journey to the asteroid, to study it, obtain samples and return back to Earth. The spacecraft is shown in Figure 2.6. It will rendezvous with the asteroid in late 2018.

As with the above missions, OSIRIS-REx will perform initial characterisation of Bennu. Optical navigation will be used to refine Bennu's ephemeris. With a closer approach, the SC will collect higher resolution images (stereophotoclinometry) to construct a shape model and identify landmarks for landing and sample collection. For the above PolyCAM, MapCam, OSIRIS-REx Laser Altimeter (OLA), OTES and OVIRS data will be used.[10] A detailed gravity field will be acquired by radiometric ranging and Doppler tracking using DSN.

The Touch and Go (TAG) operation to collect the sample and back away will also require onboard guidance and navigation. The TAGCAMs have two redundant NavCams for tracking star fields and landmarks on Bennu for SC position determination during mission operations and a StowCam. The TAG sequence is shown in Figure 2.7. NEAR and Hayabusa experiences were used to design a methodical phase approach and low-risk operations so the mission can be stopped at any point and can go back to the previous step. Consultations with ESA led to the implementation of an independent NavCam.

2.2.2. Autonomous GNC and Dual Quaternion Research

In recent years, a lot of work has been done for asteroid missions and they will be used for guidance and reference data during the thesis. There has been a growing interest in dual quaternions and their simplicity of problem representation. Dual quaternions are an upgrade to regular quaternions, since they allow the unification of translation and rotation in a single state rather than representation in separate vectors. They provide a concise, singularity free and an unambiguous rigid transform, that is also computationally minimalist. Dual quaternions hence enable robustness, efficiency, and ease of use for rigid transformation. Robotics and computer graphics are the two fields, where dual quaternions have been worked on the most. Below are the most recent works using DQs, with or without asteroid missions:

[10] http://www.asteroidmission.org/instrumentation/; date accessed: 07-10-2017

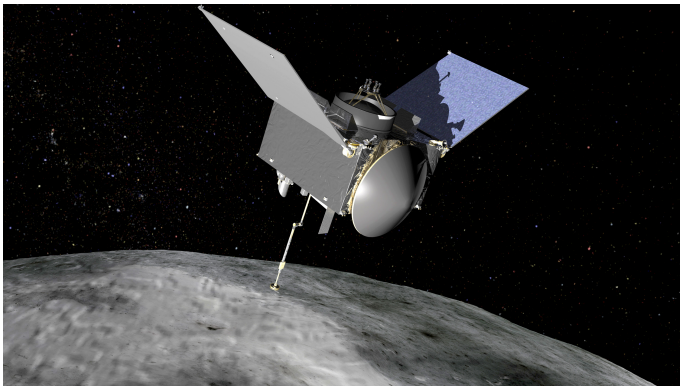


Figure 2.6: The OSIRIS-REx Spacecraft, Image Courtesy: NASA

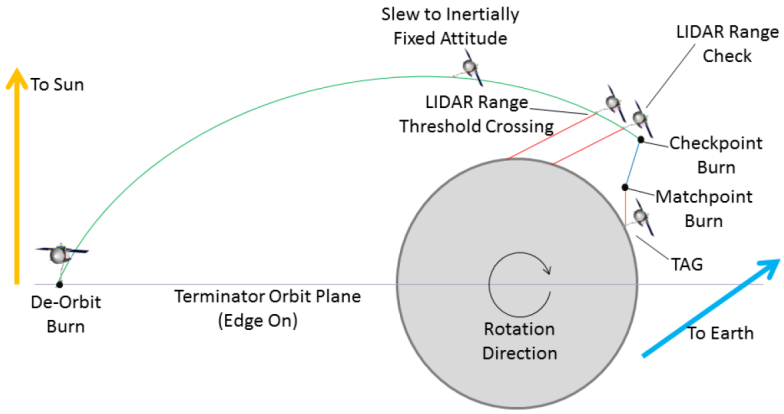


Figure 2.7: The TAG sequence of the OSIRIS-REx mission (Berry et al. (2013)).

- Zhang and Duan (2011) presented a robust finite time manoeuvre using a backstepping control strategy in the Lyapunov for a rigid SC using DQs.The settling time bound was significant in their research to be applicable for real-time application and also a certain control parameter affects both the system response and control input and therefore needs careful selection for desired control.

- Filipe and Tsiotras (2013) proposed a non-linear adaptive position and attitude tracking controller for a rigid body without the requirement of its mass and inertia matrix. Conditions placed on the reference trajectory help the identification of the mass and inertia matrix. The controller is almost asymptotically stable and is capable of handling large error angles and displacements.

- Lee and Mesbahi (2015), Lee and Mesbahi (2017) have worked with DQs for SC rendezvous with a tumbling asteroid and optimal powered descent guidance on Mars with 6 DOF LOS constraints. Their work is based on convex optimisation with piece-wise affine model predictive control (PWA-MPC). Their results show that PWA-MPC may sometimes cause the finite horizon OCP to become infeasible, i.e., the required control at a certain time-step might not be available and the computed control may not converge to the origin, i.e., the closed-loop system is not guaranteed to be asymptotically stable. hence for the stability and feasibility of PWA-MPC additional conditions may be required.

- Kwon et al. (2016) developed a virtual trajectory augmented landing control based on DQs for enabling pinpoint soft landing with a low touchdown velocity for a lunar lander. It introduces a new force-torque sequential control law to resolve the coupling of the main descent thruster and the attitude control thrusters to improve the coupled translational and rotational motion it uses the virtual landing trajectory augmented control law. Simulations show good performance and conformity with the virtual landing trajectory while the attitude is pitched up for the terminal descent.

- Dong et al. (2017) addressed integrated attitude and position control for the final phase of proximity operations of SC rendezvous and docking. They have developed a special DQ based APF to encode motion constraints of the chaser SC and a novel 6 DOF control while complying with these constraints. Stability of the system is ensured through a Lyapunov based method and the effectiveness of this method is proved by numerical simulations.

- Razgus et al. (2017) developed a relative navigation system with quaternion and dual quaternion Kalman filters (QVEKF & DQEKF). Based on the realistic simulations of the environment with typical sensing accuracies it was seen that significant gains in the estimation errors' transients were achieved by the DQEKF compared to QVEKF filter. The DQEKF filter was successful in estimating the relative position and attitude, gyroscope drift and the asteroid angular rates with high accuracy. Its error settling times were lesser by a 1000 s for asteroid rate, relative altitude and relative position and 1500 s for relative velocity than those of QVEKF filter. This results from the coupling of position and velocity using dual quaternions. A similarity between the errors' steady states was found and can be explained by the design models, that have similar expressions for linearised position error vector.

- Ballester (2018) has performed a comparison between a 6DOF controller based on classical methods and dual quaternions. It confirms the advantages of compactness of dual quaternions and similarity in their properties with attitude only controllers. The controller based on dual quaternions performed better within a certain range of enlarged perturbations, was more efficient in correcting deviations in terms of energy consumption, had shorter settling times and accumulated lesser errors for the tested scenarios. The disadvantages include higher computation times for both optimisation and integration of equation of motion.

Pavone et al. (2014) mention a list of methods to improve the GNC system to make it autonomous and applicable in real-time. Some of these methods have been implemented in recent years. There are also a few futuristic methods other than those that have been ventured and developed. We discuss the topics that are relevant and need further research below:

- Surovik and Scheeres (2014), Surovik and Scheeres (2017) have proposed a heuristic-guided objective reachability analysis for Non-Keplerian trajectory planning with a receding-horizon control or MPC. The heuristics have been formulated for efficient searching of reachable sets with a complex and finely detailed structure to manage with a scarcity of onboard computing resources. They have examined a heuristic with global search with a combination of simulated annealing and hill climbing to locate sparse regions of the planning domain. The search also satisfies the geometric and timing constraints for the remote sensing objectives for the mission whilst using receding horizon control.

- Mao et al. (2016) have presented an algorithm to solve non-convex optimal control problems via convexification in a successive manner. They have incorporated virtual control and trust regions to guarantee convergence to the optimal solution of the original problem. The advantages of this method are that every limit point in an infinite convergent case will satisfy the optimality condition for the original non-convex optimal control problem and it can be used in real-time path planning for autonomous vehicles.

- Capolupo et al. (2017) have introduced Sampling-Based Motion Planning (SBMP) and Rapidly-exploring Random Tree (RRT) algorithms for proximity operations and descent on a small Solar System body. These methods have provided an efficient exploration of the state space and have enabled construction of a feasible sequence of manoeuvres and trajectories, that abide by the constraints of the system. The advantage of these algorithms is the capacity to solve high-level planning problems and guide the SC in a cluttered environment.

- Sagliano et al. (2017) and Sagliano (2017) have treated the convex optimisation problem with pseudospectral optimal control for powered descent and landing. The flipped Radau and Lobetto families of the pseudospectral methods have been used for this research. The algorithm has been demonstrated for NASA's Mars Science Laboratory. The results show more accurate results when compared with standard transcription methods.

- Canale (2018) implemented a real-time Lyapunov-based guidance algorithm for small Solar System body proximity operations, which accounted for the oblateness of the target body causing secular effects while avoiding SC collision with the target body. This method compromises on accuracy achieved

by optimisation methods, but due to its low computational effort, it proves to be a feasible method for real-time application. The errors in right ascension of ascending node, mean anomaly and argument of periapsis, which are in the order of tens of degrees when the J_2 term[11] is ignored, were improved by two orders of magnitude by the implemented guidance algorithm.

2.3. Research Mission

With the discussion of the mission heritage and the recent research work done, we will design a research mission for this thesis. We will select the target asteroids and define the mission objectives and requirements along with the required assumptions.

2.3.1. Target Asteroids

Selection of the target asteroids needs to be done on the basis of meeting the scientific goals for the intended thesis topic. At the same time, sufficient data about the targets must be available for implementing the research on them. To evaluate the developed guidance algorithm for a perturbed environment we require asteroids with irregular shapes and varied rotation rates and sizes. In the reference mission for this thesis (Razgus (2017)), the navigation system has been tested on the asteroids Kleopatra and Itokawa and validated for the Comet 67P - Churyumov Gerasimenko. Since there is sufficient data available for these small Solar System bodies, we will use the same for this thesis. The key parameters for these asteroids are given in the Table 2.1.

Table 2.1: Key Parameters of the asteroids Kleopatra, Itokawa and comet 67P.[12]

	Kleopatra	**Itokawa**	**67P**
Semi-major axis (au)	2.79	1.32	3.46
Eccentricity	0.25	0.28	0.64
Inclination (deg)	13.11	1.62	7.04
RAAN (deg)	215.36	69.08	50.18
Argument of Perigee (deg)	180.11	162.80	12.69
Mean motion (deg/d)	0.211	0.647	0.153
Size (km)	$217 \times 94 \times 81$	$0.535 \times 0.294 \times 0.209$	$4.1 \times 3.3 \times 1.8$
Mass (kg)	4.64×10^{18}	3.51×10^{10}	9.98×10^{12}
Density (kg/m^3)	6545	1959	532.28
Rotation Period (h)	5.385	12.132	12.761
Rotation Axis $[\lambda; \beta]$ (deg)	[76; 16]	[355; -84]	[65; 59]

2.3.2. Mission Objectives

The main objective of this thesis is to enable an SC to autonomously map and observe a target asteroid and to descend for a sample collection mission. It can be broken down as below:

- **MO01** The SC shall be able to autonomously map the target body for developing in-situ maps for potential landing sites.
- **MO02** The SC shall be able to maintain the required attitude and perform manoeuvres and corrections autonomously.
- **MO03** The SC shall be able to autonomously develop a safe optimal trajectory for a touch and go descent to a particular landmark based on a priori perfect knowledge of landmarks from a global mapping phase.
- **MO04** The observational scientific instruments shall be pointed towards the target body for the observational phase of the mission for all proximity manoeuvres.

[11] J_2 term is a *zonal* term, which is the next dominating term after $-\mu/r$, while computing the gravitational potential for an oblate target body.

[12] https://ssd.jpl.nasa.gov/sbdb.cgi; date accessed: 14-02-18

2.3.3. Mission Requirements

To understand and develop the concepts required to satisfy the research objective, the mission requirements need to be defined. Using the information received from the mission heritage, we can define the requirements as given below. Detailed requirements during the different phases are discussed in Chapter 6 and Appendix A.

- **MR01** The descent phase shall involve the SC following a certain safe optimal trajectory for a particular landmark based on a priori perfect knowledge of landmarks.
  - **MR01.1** Absolute values of tolerance for position errors must be within 0.01 m in all three axes.
  - **MR01.2** Absolute values of tolerance for velocity errors must be within 0.01 m/s in all three axes.
  - **MR01.3** Absolute values of tolerance for attitude errors must be within 0.01°.
  - **MR01.4** Absolute values of tolerance for angular velocity errors must be within 0.01°/s in all three axes.
- **MR02** The mapping phase shall include high altitude, close observation and asteroid synchronous orbits in the ranges of tens of kilometres to a few hundreds of metres and the manoeuvres from one orbit to the other.
  - **MR02.1** Absolute values of tolerance for attitude errors must be within 0.01°.
  - **MR02.2** Absolute values of tolerance for angular velocity errors must be within 0.01°/s in all three axes.

2.3.4. System Requirements

From the test case scenarios used in the recent works mentioned in Subsection 2.2.2 and missions discussed in Subsection 2.2.1 we establish the system requirements as follows.

- **SR01** The SC design shall be such that it is applicable to asteroid missions to NEAs or MABs, i.e., from 0.9 au to 3.2 au approximately and that it can carry out sample collection from the asteroid surface.
- **SR02** The SC shall point towards the Sun and Earth, as required, by means of solar arrays and HGA (both with foldable joints), respectively, except during the descent phase for safety.
- **SR03** The navigation subsystem shall consist of navigation cameras, lidar, gyroscope and accelerometer to provide necessary information to the guidance system at all times.
- **SR04** The control subsystem shall have a reaction thruster for low and high thrust operations as needed for SC position.
- **SR05** The reactive thruster shall be able to provide a total nominal thrust equivalent to $m_0 \times g_0$, where m_0 and g_0 are the mass of the SC and gravitational acceleration at the initial state respectively.
- **SR06** The upper and lower limits for applicable thrust in the guidance algorithms shall be 0.95 times the maximum available thrust and 1.05 times the minimum allowable thrust, respectively.
- **SR07** The reactive thruster shall be gimballed to generate thrust within a cone of maximum angle of 20°.

2.3.5. Assumptions

For the feasibility of implementing the guidance and control algorithms, certain assumptions need to be made. They are enlisted as follows:

- **A01** The guidance problems for both descent and mapping phase are deterministic, and the stochastic influences from navigation errors are negligible, i.e., ideal/perfect navigation.
- **A02** The ephemerides data, polyhedron gravity model and axial tilt of the target body are available.
- **A03** The small body rotates about itself with a constant angular velocity with known uncertainties.
- **A04** A priori global maps are available for navigation and landmark identification for the descent and therefore no image processing is needed.

- **A05** The SC is considered to have been stably injected into any required orbit and that no tumbling is present before mapping or descent phase.
- **A06** The control actuator does not generate mass expulsion disturbance torques.
- **A07** The SC solar arrays are assumed to have steerable and foldable capabilities, such that a touch and go sampling descent is possible.
- **A08** The thermal recoil forces on the SC are very small compared to other perturbing forces, and hence will not be considered.
- **A09** The Sun angle with the asteroid will be assumed to be constant for the period of mapping and descent each.

2.3.6. Spacecraft

The SC chosen for the mission objective of mapping and descent is Rosetta. This SC is ideal for the phase of mapping, at distances between ≈1.2 to 3 au, but not for descent. Hence, we assume in **A07** that the SC solar arrays have foldable capabilities and do not require Sun pointing during the descent phase. The solar arrays would be similar to OSIRIS-REx for the descent phase. Simplified polyhedron models for the SC is presented in Section 7.2, for mapping and descent. In actual applications, the concept of Roll-Out Solar Arrays, which is at present state-of-the-art technology for solar panels, would apply for such a mission that involves mapping and descent phases at such large distances. A successful implementation of this technology has already been implemented onboard the International Space Station (ISS).[13] Link to the video for the interested reader has been provided here.

Table 2.2: Specifications of Rosetta and OSIRIS-REx as reference SCs for mapping and descent.[14]

	Rosetta	**OSIRIS-REx**
Launch Mass (kg)	2900	2110
Dry mass (kg)	1230	880
Thrusters (N)	10 (24)	222 (4), 22 (6), 4.4 (16), 0.44 (2)
Mass flow rate (kg/s)	0.0035	-
AOCS	Three-axis stabilised	Three-axis stabilised
Power Source	Solar Arrays and Batteries	Solar Arrays and Batteries
SC Dimensions (m)	2.8 × 2.1 × 2	2.44 × 2.44 × 3.15
Solar Array Dimensions (m)	14 × 3 (2)	1.9 × 2.3 (2)
Sample Collector (m)	-	3.5

2.4. Conclusion

We discussed the asteroid classes and properties in this chapter along with past and current missions to such small Solar System bodies. We also discussed some of the most relevant recent works in the development of GNC algorithms for better computational efficiency in autonomous applications. Based on these, we selected the asteroids Kleopatra, Itokawa and comet 67P for implementation in the guidance algorithms, developed during this thesis. From the advantages seen in the previous research work guidance algorithms based on successive convexification and motion planning will be developed for the descent and mapping phases respectively. These algorithms will be developed with quaternions and dual quaternions and compared for their advantages and disadvantages.

[13] https://www.nasa.gov/mission_pages/station/research/experiments/2139.html, date accessed:01-03-19
[14] http://spaceflight101.com/osiris-rex/osiris-rex-spacecraft-overview/, date accessed:01-03-2019

3

Dual Quaternions

Rigid body orientations and translations can be represented by a number of methods, of which the most popular ones are design cosine matrices (DCMs), Euler angles, Euler-eigenaxis rotation, quaternions and Rodrigues parameters. The advantages of quaternions and modified Rodrigues parameters weigh over the rest and can be found in detail in Markley and Crassidis (2013) for reference. Dual quaternions are an extension to quaternions and provide a tool to help express and analyse physical properties of rigid bodies. They are eight-dimensional numbers, that help represent translation and rotation in a single state. They thereby cut down the volume of algebra and make the problem more concise as well as robust and singularity free. We shall introduce dual quaternions and how they can be used to represent rigid transformations in this chapter.

3.1. Dual Numbers

Dual numbers were introduced in 1873, by William Clifford as an extension to real numbers.[1] With the concept of complex numbers adding the imaginary number i, with the property, that $i^2 = -1$, he brought forth the element ϵ, which has the property of being nilpotent, that is $\epsilon^2 = 0$ although $\epsilon \neq 0$. The form of a dual number is analogous to complex numbers and is written as,

$$\check{d} = a + b\epsilon \tag{3.1}$$

where, $a, b \in \mathbb{R}$ and '$\check{d}$' represents dual nature of a number and will be used throughout the literature. The number when introduced was used to represent dual angles, which measured the relative position of two skewed lines in three-dimensional space. In d, a represented the angle between the directions of the two lines and b the distance between them. It is actually a special case of one-dimensional algebra later generalised to n dimensions and known as Grassman numbers, which are now used as a foundation to define superspace or concepts like supersymmetry. Understanding dual numbers are key to understanding dual quaternions. In Eq. (3.1) a represents the real part of the number and b the dual part.

3.1.1. Dual Numbers Arithmetic Operations

Addition, subtraction, multiplication and division between dual numbers are shown below (Jia (2013)).

$$\check{d}_1 \pm \check{d}_2 = (a_1 \pm a_2) + (b_1 \pm b_2)\epsilon \tag{3.2}$$

$$\check{d}_1 \check{\otimes} \check{d}_2 = a_1 a_2 + (a_1 b_2 + b_1 a_2)\epsilon \tag{3.3}$$

$$\frac{\check{d}_1}{\check{d}_2} = \frac{a_1 + b_1\epsilon}{a_2 + b_2\epsilon} = \frac{a_1 + b_1\epsilon}{a_2 + b_2\epsilon} \cdot \frac{a_2 - b_2\epsilon}{a_2 - b_2\epsilon} = \frac{a_1 a_2 + (b_1 a_2 - a_1 b_2)\epsilon}{a_2^2} \tag{3.4}$$

The inverse of a dual number exists as long as $a \neq 0$, given as,

$$\check{d}^{-1} = a^{-1}(1 - ba^{-1}\epsilon) \tag{3.5}$$

[1] https://en.wikipedia.org/wiki/Dual_number; date accessed: 15-10-2017

A function of a dual number can be defined using Taylor series as,

$$\begin{aligned} f(a+b\epsilon) &= f(a) + \frac{f'(a)}{1!}b\epsilon + \frac{f''(a)}{2!}b^2\epsilon^2 + ... \\ &= f(a) + bf'(a)\epsilon \end{aligned} \tag{3.6}$$

since $\epsilon^2 = 0$ and any higher power of ϵ would be zero. This equation enables to establish a number of functions of dual numbers. The derivative also helps find the tangent to an arbitrary point on a parametric curve, which is equal to the normalised dual part of the point (Kenwright (2012)).

3.1.2. Dual Vectors

A dual vector, $\check{\boldsymbol{d}} = (\check{d}_1,\ \check{d}_2,\ \check{d}_3)$ consists of three dual numbers $\check{d}_1$, $\check{d}_2$ and $\check{d}_3$. The product of a dual vector with a dual number is another dual vector, given as follows (Jia (2013)),

$$\check{d}\check{\boldsymbol{d}} = (\check{d} \otimes \check{d}_1,\ \check{d} \otimes \check{d}_2,\ \check{d} \otimes \check{d}_3) \tag{3.7}$$

The inner and outer products of two dual vectors $\check{\boldsymbol{d}}$ and $\check{\boldsymbol{e}}$ are similar to those of two normal vectors and are as below (Jia (2013)),

$$\check{\boldsymbol{d}} \cdot \check{\boldsymbol{e}} = \check{d}_1 \otimes \check{e}_1 + \check{d}_2 \otimes \check{e}_2 + \check{d}_3 \otimes \check{e}_3 \tag{3.8}$$

$$\check{\boldsymbol{d}} \times \check{\boldsymbol{e}} = \begin{pmatrix} \check{d}_2 \otimes \check{e}_3 - \check{d}_3 \otimes \check{e}_2 \\ \check{d}_3 \otimes \check{e}_1 - \check{d}_1 \otimes \check{e}_3 \\ \check{d}_1 \otimes \check{e}_2 - \check{d}_2 \otimes \check{e}_1 \end{pmatrix}^T \tag{3.9}$$

These equations will be used ahead for the algebraic operations of dual quaternions.

3.2. Dual Quaternions

Dual Quaternions (DQs) are an extension of dual numbers and quaternions together. They can be used to solve a range of complex problems in a compact and elegant fashion. A rigid transformation is one such problem, that can be tackled with dual quaternions. DQs have eight scalar variables, that are combined through concatenation (Jia (2013)). It helps to achieve smooth interpolation between rigid transforms seamlessly and effortlessly. A dual quaternion is of the form:

$$\check{\boldsymbol{q}} = \boldsymbol{q}_r + \boldsymbol{q}_d\epsilon \tag{3.10}$$

where,

$$\boldsymbol{q}_r = q_{r_1}i + q_{r_2}j + q_{r_3}k + q_{r_4} \tag{3.11}$$

$$\boldsymbol{q}_d = q_{d_1}i + q_{d_2}j + q_{d_3}k + q_{d_4} \tag{3.12}$$

where, i, j and k are imaginary numbers and the quaternions, $\boldsymbol{q}_r$ & $\boldsymbol{q}_d$, are defined by the properties in Eqs. (3.10), (3.11) and (3.12). As per the definition of dual numbers, $\boldsymbol{q_r}$ represents the real part and $\boldsymbol{q_d}$ the dual part. It can also be represented as the sum of a dual number and a dual vector, as,

$$\check{\boldsymbol{q}} = (q_{r_4} + q_{d_4}\epsilon) + (\boldsymbol{q_{r_{1:3}}} + \boldsymbol{q_{d_{1:3}}}\epsilon) \tag{3.13}$$

The DQ can also be written as an 8-tuple, as, vector(Jia (2013))

$$\check{\boldsymbol{q}} = (\boldsymbol{q_r}\ \boldsymbol{q_d})^T = (q_{r_1},\ q_{r_2},\ q_{r_3},\ q_{r_4},\ q_{d_1},\ q_{d_2},\ q_{d_3},\ q_{d_4})^T \tag{3.14}$$

This representation makes the dual quaternion algebraic operations easy to handle and we will use it in the literature study henceforth in this form.

3.2.1. Dual Quaternions Geometric Algebra

The addition and subtraction of DQs can be generalised from the operations of dual numbers, as,

$$\check{\boldsymbol{q}}_1 \pm \check{\boldsymbol{q}}_2 = (\boldsymbol{q_{r_1}} \pm \boldsymbol{q_{r_2}}) + (\boldsymbol{q_{d_1}} \pm \boldsymbol{q_{d_2}})\epsilon \tag{3.15}$$

In a similar fashion as Eq. (3.3), for dual numbers, the multiplication of two DQs can be given as,

$$\check{\boldsymbol{q}}_1 \check{\otimes} \check{\boldsymbol{q}}_2 = \boldsymbol{q}_{\boldsymbol{r_1}} \otimes \boldsymbol{q}_{\boldsymbol{r_2}} + (\boldsymbol{q}_{\boldsymbol{r_1}} \otimes \boldsymbol{q}_{\boldsymbol{d_2}} + \boldsymbol{q}_{\boldsymbol{d_1}} \otimes \boldsymbol{q}_{\boldsymbol{r_2}})\epsilon \tag{3.16}$$

$$\check{\boldsymbol{q}}_2 \check{\odot} \check{\boldsymbol{q}}_1 = \boldsymbol{q}_{\boldsymbol{r_2}} \odot \boldsymbol{q}_{\boldsymbol{r_1}} + (\boldsymbol{q}_{\boldsymbol{d_2}} \odot \boldsymbol{q}_{\boldsymbol{r_1}} + \boldsymbol{q}_{\boldsymbol{d_1}} \odot \boldsymbol{q}_{\boldsymbol{r_2}})\epsilon \tag{3.17}$$

Using the quaternion product property,

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \boldsymbol{q}_2 \odot \boldsymbol{q}_1 \tag{3.18}$$

we can see, that it also applies to the dual quaternion products, as shown below.

$$\check{\boldsymbol{q}}_1 \check{\otimes} \check{\boldsymbol{q}}_2 = \check{\boldsymbol{q}}_2 \check{\odot} \check{\boldsymbol{q}}_1 \tag{3.19}$$

We can represent Eqs. (3.16) and (3.17) in a matrix form, as,

$$\check{\boldsymbol{q}}_1 \check{\otimes} \check{\boldsymbol{q}}_2 = \begin{bmatrix} [\boldsymbol{q}_{\boldsymbol{r_1}} \otimes] & 0_{4\times4} \\ [\boldsymbol{q}_{\boldsymbol{d_1}} \otimes] & [\boldsymbol{q}_{\boldsymbol{r_1}} \otimes] \end{bmatrix} \begin{pmatrix} \boldsymbol{q}_{\boldsymbol{r_2}} \\ \boldsymbol{q}_{\boldsymbol{d_2}} \end{pmatrix} \tag{3.20}$$

$$\check{\boldsymbol{q}}_2 \check{\odot} \check{\boldsymbol{q}}_1 = \begin{bmatrix} [\boldsymbol{q}_{\boldsymbol{r_2}} \odot] & 0_{4\times4} \\ [\boldsymbol{q}_{\boldsymbol{d_2}} \odot] & [\boldsymbol{q}_{\boldsymbol{r_2}} \odot] \end{bmatrix} \begin{pmatrix} \boldsymbol{q}_{\boldsymbol{r_1}} \\ \boldsymbol{q}_{\boldsymbol{d_1}} \end{pmatrix} \tag{3.21}$$

An inverse of the dual quaternion exists if $\check{\boldsymbol{q}}_{\boldsymbol{r}} \neq 0$ and is given as,

$$\check{\boldsymbol{q}}^{-1} = {\boldsymbol{q}_{\boldsymbol{r}}}^{-1}(1 - \boldsymbol{q}_{\boldsymbol{d}} {\boldsymbol{q}_{\boldsymbol{r}}}^{-1}\epsilon) \tag{3.22}$$

3.2.2. Dual Quaternion Conjugates

DQs can have three possible conjugates, since they have two quaternion elements.

$$\check{\boldsymbol{q}}^{\bullet} = \boldsymbol{q}_{\boldsymbol{r}} - \boldsymbol{q}_{\boldsymbol{d}}\epsilon \tag{3.23}$$

$$\check{\boldsymbol{q}}^{*} = \boldsymbol{q}_{\boldsymbol{r}}^{*} + \boldsymbol{q}_{\boldsymbol{d}}^{*}\epsilon \tag{3.24}$$

$$\check{\boldsymbol{q}}^{\diamond} = \boldsymbol{q}_{\boldsymbol{r}}^{*} - \boldsymbol{q}_{\boldsymbol{d}}^{*}\epsilon \tag{3.25}$$

Conjugates of the above conjugates in the same fashion gives the DQ itself. As 8-tuples, the above conjugates are represented in the same order, as,

$$\begin{aligned}
\check{\boldsymbol{q}}^{\bullet} &= (q_{r_1},\, q_{r_2},\, q_{r_3},\, q_{r_4},\, -q_{d_1},\, -q_{d_2},\, -q_{d_3},\, -q_{d_4})^T \\
\check{\boldsymbol{q}}^{*} &= (-q_{r_1},\, -q_{r_2},\, -q_{r_3},\, q_{r_4},\, -q_{d_1},\, -q_{d_2},\, -q_{d_3},\, q_{d_4})^T \\
\check{\boldsymbol{q}}^{\diamond} &= (-q_{r_1},\, -q_{r_2},\, -q_{r_3},\, q_{r_4},\, q_{d_1},\, q_{d_2},\, q_{d_3},\, -q_{d_4})^T
\end{aligned}$$

Multiplication of the DQ with its conjugates need to be examined in order to realise which conjugate would prove to be useful.

$$\begin{aligned}
\check{\boldsymbol{q}} \check{\otimes} \check{\boldsymbol{q}}^{\bullet} &= (\boldsymbol{q}_{\boldsymbol{r}} + \boldsymbol{q}_{\boldsymbol{d}}\epsilon)(\boldsymbol{q}_{\boldsymbol{r}} - \boldsymbol{q}_{\boldsymbol{d}}\epsilon) \\
&= \boldsymbol{q}_{\boldsymbol{r}} \otimes \boldsymbol{q}_{\boldsymbol{r}} + (\boldsymbol{q}_{\boldsymbol{d}} \otimes \boldsymbol{q}_{\boldsymbol{r}} - \boldsymbol{q}_{\boldsymbol{r}} \otimes \boldsymbol{q}_{\boldsymbol{d}})\epsilon
\end{aligned} \tag{3.26}$$

All the cross products in this product cancel out. Since, $\boldsymbol{q}_{\boldsymbol{r}} \otimes \boldsymbol{q}_{\boldsymbol{r}}$ is not scalar and $\boldsymbol{q}_{\boldsymbol{d}} \otimes \boldsymbol{q}_{\boldsymbol{r}} \neq \boldsymbol{q}_{\boldsymbol{r}} \otimes \boldsymbol{q}_{\boldsymbol{d}}$, this conjugate multiplication is not really useful and we move to the product of the next conjugate with the DQ.

$$\begin{aligned}
\check{\boldsymbol{q}} \check{\otimes} \check{\boldsymbol{q}}^{*} &= (\boldsymbol{q}_{\boldsymbol{r}} + \boldsymbol{q}_{\boldsymbol{d}}\epsilon)(\boldsymbol{q}_{\boldsymbol{r}}^{*} + \boldsymbol{q}_{\boldsymbol{d}}^{*}\epsilon) \\
&= \boldsymbol{q}_{\boldsymbol{r}} \otimes \boldsymbol{q}_{\boldsymbol{r}}^{*} + (\boldsymbol{q}_{\boldsymbol{r}} \otimes \boldsymbol{q}_{\boldsymbol{d}}^{*} + \boldsymbol{q}_{\boldsymbol{d}} \otimes \boldsymbol{q}_{\boldsymbol{r}}^{*})\epsilon
\end{aligned} \tag{3.27}$$

In this case $\boldsymbol{q}_{\boldsymbol{r}} \otimes {\boldsymbol{q}_{\boldsymbol{r}}}^{*}$ is scalar as seen from quaternion multiplication. The dual part of the product also comes out to be scalar once it is expanded and the quaternion products are established.

$$\begin{aligned}
\check{\boldsymbol{q}} \check{\otimes} \check{\boldsymbol{q}}^{*} &= \|\boldsymbol{q}_r\|^2 + 2(q_{r_1} q_{d_1} + q_{r_2} q_{d_2} + q_{r_3} q_{d_3} + q_{r_4} q_{d_4})\epsilon \\
&= \|\boldsymbol{q}_r\|^2 + 2(q_{r_4} q_{d_4} + \boldsymbol{q}_{\boldsymbol{r}} \cdot \boldsymbol{q}_{\boldsymbol{d}})\epsilon
\end{aligned} \tag{3.28}$$

This shows, that this product is nothing but a dual number. If $\boldsymbol{q}_{\boldsymbol{r}}$ and $\boldsymbol{q}_{\boldsymbol{d}}$, i.e., the real and dual part of the DQ, are orthogonal to each other the dual part of the above result equates to to zero. In such a case the

multiplication of the DQ and conjugate will be truly scalar. Moving on to the last kind of product of a DQ and its conjugate,

$$\begin{aligned}\check{\boldsymbol{q}}\check{\otimes}\check{\boldsymbol{q}}^{\diamond} &= (\boldsymbol{q_r} + \boldsymbol{q_d}\epsilon)(\boldsymbol{q_r^*} - \boldsymbol{q_d^*}\epsilon) \\ &= \boldsymbol{q_r} \otimes \boldsymbol{q_r^*} + (\boldsymbol{q_d} \otimes \boldsymbol{q_r^*} - \boldsymbol{q_r} \otimes \boldsymbol{q_d^*})\epsilon \\ &= \|\boldsymbol{q}_r\|^2 + 2(q_{r_4}\boldsymbol{q_{d_{1:3}}} - q_{d_4}\boldsymbol{q_{r_{1:3}}} + \boldsymbol{q_{r_{1:3}}} \times \boldsymbol{q_{d_{1:3}}})\epsilon\end{aligned} \tag{3.29}$$

This results in a dual quaternion, since even though the real part is scalar, it could represent a real quaternion and the dual part represents a pure quaternion.

The conjugates of the the product of two DQs can be proved by mathematical treatment to be the product of the individual conjugates in reverse order.

$$\begin{aligned}(\check{\check{\boldsymbol{q}}}\check{\otimes}\check{\boldsymbol{q}})^{\bullet} &= \check{\boldsymbol{q}}^{\bullet}\check{\otimes}\check{\check{\boldsymbol{q}}}^{\bullet} \\ (\check{\check{\boldsymbol{q}}}\check{\otimes}\check{\boldsymbol{q}})^{*} &= \check{\boldsymbol{q}}^{*}\check{\otimes}\check{\check{\boldsymbol{q}}}^{*} \\ (\check{\check{\boldsymbol{q}}}\check{\otimes}\check{\boldsymbol{q}})^{\diamond} &= \check{\boldsymbol{q}}^{\diamond}\check{\otimes}\check{\check{\boldsymbol{q}}}^{\diamond}\end{aligned}$$

3.2.3. Unit Dual Quaternion

A DQ can be called a unit DQ if it satisfies the condition, that its product with the second conjugate is 1.

$$\check{\boldsymbol{q}}\check{\otimes}\check{\boldsymbol{q}}^{*} = 1 \tag{3.30}$$

which implies,

$$q_{r_1}^2 + q_{r_2}^2 + q_{r_3}^2 + q_{r_4}^2 = 1 \tag{3.31}$$

$$q_{r_1}q_{d_1} + q_{r_2}q_{d_2} + q_{r_3}q_{d_3} + q_{r_4}q_{d_4} = 0 \tag{3.32}$$

The real part of the DQ must be a unit quaternion and it must be orthogonal to the dual part (quaternion) of the DQ. This also leads to the inverse of the DQ being equal to its conjugate.

$$\check{\boldsymbol{q}}^{-1} = \check{\boldsymbol{q}}^{*} \tag{3.33}$$

The consequence of the conditions in Eqs. (3.31) and (3.32) are that the DOFs are reduced from eight to six, as it should be exactly for a rigid body in 3D.

3.3. Dual Quaternion Attitude and Position Representation

We have discussed all the properties of dual numbers and dual quaternions in the above section. These will be used in this section to establish how they can be used to represent the attitude and position of an SC with respect to a reference frame.

3.3.1. Rigid Displacement

Considering a vector $\boldsymbol{r}$ rotating and translating through 3D space. The rotation can be taken care of by a rotation matrix C and the translation is equivalent to the vector $\boldsymbol{t}$. The vector after displacement can be therefore given by, $\boldsymbol{r}' = C\boldsymbol{r} + \boldsymbol{t}$. Using dual quaternions both these motions can be represented by a dual quaternion. This makes the representation compact and results in easier computation.

The real part and dual part of the DQ are both quaternions. If the dual part is zero, i.e., $\check{\boldsymbol{q}} = \boldsymbol{q_r} + 0\,\epsilon$ and $\boldsymbol{q_r}$ is a unit quaternion, the DQ enables 3D rotation with the unit quaternion. This can be proved by using a unit quaternion $\check{\boldsymbol{p}}$ representing a vector, that is to be rotated about the DQ $\check{\boldsymbol{q}}$.

$$\begin{aligned}\check{\boldsymbol{q}}\check{\otimes}\check{\boldsymbol{p}}\check{\otimes}\check{\boldsymbol{q}}^{\diamond} &= \boldsymbol{q_r}(1 + (p_1\ i + p_2\ j + p_3\ k)\ \epsilon)\boldsymbol{q_r^*} \\ &= \boldsymbol{q_r}\boldsymbol{q_r^*} + \boldsymbol{q_r}(p_1\ i + p_2\ j + p_3\ k)\ \epsilon\ \boldsymbol{q_r^*} \\ &= 1 + \boldsymbol{q_r}(p_1\ i + p_2\ j + p_3\ k)\ \epsilon\ \boldsymbol{q_r^*}\end{aligned} \tag{3.34}$$

The second term can be seen as the operation used in quaternions for rotation and hence it can be proved, that a DQ can be used to rotate a vector in 3D. For translation, a unit DQ $\check{\boldsymbol{q}} = 1 + \boldsymbol{t}/2\,\epsilon$ whose dual part is given by a pure quaternion $\boldsymbol{t}$ representing the vector of translation can be used to achieve the required motion.

In analogy with quaternions, that use half angles for rotation, half of the translation quaternion is used for complete translation (Kavan et al. (2007)). Using the same expression as earlier

$$\begin{aligned}\check{\boldsymbol{q}}\check{\otimes}\check{\boldsymbol{p}}\check{\otimes}\check{\boldsymbol{q}}^{\diamond} &= (1+\frac{\boldsymbol{t}}{2}\,\epsilon)(1+(p_1\;i+p_2\;j+p_3k)\,\epsilon)\;(1-\frac{\boldsymbol{t}^{\,*}}{2}\;\epsilon)\\ &= (1+\frac{\boldsymbol{t}}{2}\,\epsilon)(1-\frac{\boldsymbol{t}^{\,*}}{2}\;\epsilon)+(1+\frac{\boldsymbol{t}}{2}\,\epsilon)\;\boldsymbol{p}\,\epsilon\;(1-\frac{\boldsymbol{t}^{\,*}}{2}\;\epsilon)\\ &= 1+\boldsymbol{t}\,\epsilon+\boldsymbol{p}\,\epsilon\\ &= 1+((p_1+t_1)i+(p_2+t_2)j+(p_3+t_3)k)\,\epsilon\end{aligned} \tag{3.35}$$

which proves, that a unit dual quaternion can be used for just translation. Combining the unit dual quaternions for rotation and translation for a single dual quaternion to represent both motions at once,

$$\check{\boldsymbol{q}} = (1+\frac{\boldsymbol{t}}{2}\,\epsilon)\boldsymbol{q_r} = \boldsymbol{q_r}+\frac{\boldsymbol{t}}{2}\boldsymbol{q_r}\,\epsilon \tag{3.36}$$

$$\check{\boldsymbol{q}} = \boldsymbol{q_r}+\boldsymbol{q_d}\,\epsilon \tag{3.37}$$

As seen in the section of quaternions, unit quaternions for rotation can be represented using half angles and for translation a vector in Cartesian coordinates.

$$\boldsymbol{q_r} = \cos\frac{\theta}{2}+\hat{\boldsymbol{n}}\sin\frac{\theta}{2} \tag{3.38}$$

$$\boldsymbol{t} = t_1\;i+t_2\;j+t_3\;k \tag{3.39}$$

Using the above quaternions in the DQ established in Eq. (3.37)

$$\begin{aligned}\check{\boldsymbol{q}} &= \boldsymbol{q_r}+(-\boldsymbol{t}\cdot\boldsymbol{q_r}+q_{r_4}\boldsymbol{t}+\boldsymbol{t}\times\boldsymbol{q_r})\,\frac{\epsilon}{2}\\ \check{\boldsymbol{q}} &= \cos\frac{\theta}{2}+\hat{\boldsymbol{n}}\sin\frac{\theta}{2}+\left(-\sin\frac{\theta}{2}(\boldsymbol{t}\cdot\hat{n})+\cos\frac{\theta}{2}\boldsymbol{t}+\sin\frac{\theta}{2}(\boldsymbol{t}\times\hat{n})\right)\frac{\epsilon}{2}\end{aligned} \tag{3.40}$$

It can be easily proved, that the DQ representing both motions is itself a unit quaternion.

$$\begin{aligned}\check{\boldsymbol{q}}\check{\otimes}\check{\boldsymbol{q}}^{*} &= \left(\boldsymbol{q_r}+\boldsymbol{t}\boldsymbol{q_r}\,\frac{\epsilon}{2}\right)\left(\boldsymbol{q_r}^{*}+(\boldsymbol{t}\boldsymbol{q_r})^{*}\,\frac{\epsilon}{2}\right)\\ &= \boldsymbol{q_r}\boldsymbol{q_r}^{*}+\left(\boldsymbol{q_r}(\boldsymbol{t}\boldsymbol{q_r})^{*}+\boldsymbol{t}\boldsymbol{q_r}\boldsymbol{q_r}^{*}\right)\frac{\epsilon}{2}\\ &= 1+\left(\boldsymbol{q_r}\boldsymbol{q_r}^{*}\boldsymbol{t}^{*}+\boldsymbol{t}\right)\frac{\epsilon}{2}\\ &= 1+\left(\boldsymbol{t}^{*}+\boldsymbol{t}\right)\frac{\epsilon}{2}\\ &= 1\end{aligned} \tag{3.41}$$

As discussed in Section 3.2.3, the above product being 1 is the condition for the DQ to be a unit DQ and also the property $\check{\boldsymbol{q}}^{-1}=\check{\boldsymbol{q}}^{*}$ is applicable for the unit DQ. With the above generalisations, the rotation and translation unit quaternions can be figured out from any unit DQ.

Earlier in this section we proved the use of separate unit DQs to rotate and translate a vector. We use the same operation with the combined unit DQ and the vector to get the final DQ after motion.

$$\begin{aligned}\check{\boldsymbol{q}}\check{\otimes}\check{\boldsymbol{p}}\check{\otimes}\check{\boldsymbol{q}}^{\diamond} &= \left(\boldsymbol{q_r}+\boldsymbol{t}\boldsymbol{q_r}\frac{\epsilon}{2}\right)(1+\boldsymbol{p}\,\epsilon)\left(\boldsymbol{q_r}^{*}-(\boldsymbol{t}\boldsymbol{q_r})^{*}\frac{\epsilon}{2}\right)\\ &= \boldsymbol{q_r}\boldsymbol{q_r}+(\boldsymbol{q_r}\boldsymbol{p}\boldsymbol{q_r}^{*}+\frac{1}{2}\boldsymbol{t}\boldsymbol{q_r}\boldsymbol{q_r}^{*}-\frac{1}{2}\boldsymbol{q_r}\boldsymbol{q_r}^{*}\boldsymbol{t}^{*})\,\epsilon\\ &= 1+(\boldsymbol{q_r}\boldsymbol{p}\boldsymbol{q_r}^{*}+\boldsymbol{t})\,\epsilon\end{aligned} \tag{3.42}$$

The above transformation takes care of rotation first and then translation. In case translation is done before rotation the DQ $\check{\boldsymbol{q}}=\boldsymbol{q_r}+\boldsymbol{q_r}\frac{\boldsymbol{t}}{2}\,\epsilon$ should be used in the same operation as above, which gives the result,

$$\check{\boldsymbol{q}}\check{\otimes}\check{\boldsymbol{p}}\check{\otimes}\check{\boldsymbol{q}}^{\diamond} = 1+\boldsymbol{q_r}(\boldsymbol{p}+\boldsymbol{t})\boldsymbol{q_r}^{*}\,\epsilon \tag{3.43}$$

For a sequence of transformations on the vector $\boldsymbol{p}$,

$$\boldsymbol{p}' = \check{\boldsymbol{q}}_n \check{\otimes} (\check{\boldsymbol{q}}_{n-1} \check{\otimes} \ldots (\check{\boldsymbol{q}}_1 \check{\otimes} \check{\boldsymbol{p}} \check{\otimes} \check{\boldsymbol{q}}_1^{\diamond}) \ldots \check{\otimes} \check{\boldsymbol{q}}_{n-1}^{\diamond}) \check{\otimes} \check{\boldsymbol{q}}_n^{\diamond} \tag{3.44}$$

A single DQ representing all the transformations as $\check{\boldsymbol{Q}} = \check{\boldsymbol{q}}_n \check{\otimes} \check{\boldsymbol{q}}_{n-1} \check{\otimes} \ldots \check{\otimes} \check{\boldsymbol{q}}_2 \check{\boldsymbol{q}}_1$ is used and hence,

$$\boldsymbol{p}' = \check{\boldsymbol{Q}} \check{\otimes} \check{\boldsymbol{p}} \check{\otimes} \check{\boldsymbol{Q}}^{\diamond} \tag{3.45}$$

Differentiation

For a DQ, differentiation is carried out separately on the real and dual parts, as,

$$\dot{\check{\boldsymbol{q}}} = \dot{\boldsymbol{q}}_r + \frac{1}{2}(\dot{\boldsymbol{t}} \boldsymbol{q}_r + \boldsymbol{t} \dot{\boldsymbol{q}}_r) \tag{3.46}$$

where, $\dot{\boldsymbol{q}}_r = \frac{1}{2}\boldsymbol{\omega} \boldsymbol{q}_r$, $\boldsymbol{\omega}$ being the angular velocity vector and hence a pure quaternion. $\dot{\boldsymbol{t}} = \boldsymbol{v}$, where $\boldsymbol{v}$ is the velocity vector and therefore, a pure quaternion. The differentiation of DQ thus becomes,

$$\dot{\check{\boldsymbol{q}}} = \frac{1}{2}\boldsymbol{\omega} \boldsymbol{q}_r + \left(\frac{1}{2}\boldsymbol{v} \boldsymbol{q}_r + \frac{1}{4}\boldsymbol{t}\boldsymbol{\omega} \boldsymbol{q}_r\right)\epsilon \tag{3.47}$$

3.4. Screw Motion with Dual Quaternions

In the previous sections, we have discussed how a unit DQ can be used for the representation of the rotational and translational motion of a vector. In this section, we will discuss the rigid transformations of lines using DQs. Any rigid displacement is equivalent to screw motion, i.e., when the rotation is taking place about an axis and/or the object is moving along the axis.

3.4.1. Plücker Coordinates

Plücker coordinates were introduced in geometry in the 19^{th} century by Julius Plücker to assign six homogeneous coordinates to each line in a prospective 3D Euclidean space. In 3D space, a line is determined by the coordinates of two points on the line or by two planes, that contain the line. A vector displacement from one distinct point to another distinct point on the line would represent the direction of the line and its magnitude would be nonzero. If a particle moves along this line, it would have a moment about the origin of the reference frame being used.

In Figure 3.1, $\boldsymbol{p}$ is the position vector of a point on the line with respect to the reference frame and $\boldsymbol{l}$ be the vector representing the direction of the line. The momentum of an object moving along the line can then be given by a momentum vector, $\boldsymbol{m} = \boldsymbol{p} \times \boldsymbol{l}$. The Plücker coordinates are then given as ($\boldsymbol{l}$, $\boldsymbol{m}$). The coordinate $\boldsymbol{m}$ is independent of the point $\boldsymbol{p}$, since any point on the line will give the same moment vector. Consider the point $\boldsymbol{p}'$ on the Figure 3.1. Since it is on the same line as $\boldsymbol{p}$, it satisfies the condition $\boldsymbol{p}' - \boldsymbol{p} = \lambda \boldsymbol{l}$ for a certain value of λ. We can prove the independency of $\boldsymbol{m}$ of the point on the line as follows,

$$\begin{aligned} \boldsymbol{p}' \times \boldsymbol{l} &= (\boldsymbol{p} + (\boldsymbol{p}' - \boldsymbol{p})) \times \boldsymbol{l} \\ &= \boldsymbol{p} \times \boldsymbol{l} + (\boldsymbol{p}' - \boldsymbol{p}) \times \boldsymbol{l} \\ &= \boldsymbol{m} + \lambda \boldsymbol{l} \times \boldsymbol{l} \\ &= \boldsymbol{m} \end{aligned} \tag{3.48}$$

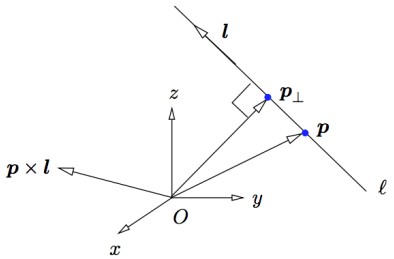


Figure 3.1: Plücker coordinates (Jia (2013))

The distance of the line from the origin can simply be given by $\|m\|/\|l\|$.These coordinates are also known as homogeneous coordinates, because they are scalable, which means, that ($\boldsymbol{l}$, $\boldsymbol{m}$) and ($c\boldsymbol{l}$, $c\boldsymbol{m}$) represent the same line.

When two particular constraints are applied, two variables are lost from the coordinates making the line having four DOFs in space. One of them being, that the line is a unit vector ($\|\boldsymbol{l}\| = 1$) and the second being the dot product of the line and moment vector equates to zero ($\boldsymbol{l} \cdot \boldsymbol{m} = \boldsymbol{l}.(\boldsymbol{p} \times \boldsymbol{l}) = 0$). Under these conditions, the moment vector becomes the moment of the line due to a unit force acting on point $\boldsymbol{p}$ on the line in the direction of the unit vector $\hat{\boldsymbol{l}}$ with respect to the origin. Also the magnitude of the moment vector becomes the distance to closest point on the line from the origin. Referring to Figure 3.2, we see,

$$\begin{aligned} \boldsymbol{p}_\perp &= \boldsymbol{p} - \boldsymbol{p}\cos\theta \\ &= (\hat{\boldsymbol{l}} \cdot \hat{\boldsymbol{l}}) - (\hat{\boldsymbol{l}} \cdot \boldsymbol{p})\hat{\boldsymbol{l}} \\ &= \hat{\boldsymbol{l}} \times (\boldsymbol{p} \times \hat{\boldsymbol{l}}) \\ &= \hat{\boldsymbol{l}} \times \boldsymbol{m} \\ &= \boldsymbol{m} \end{aligned} \tag{3.49}$$

Therefore, $\|\boldsymbol{p}_\perp\| = \|m\|$ for a line represented by a unit vector $\hat{\boldsymbol{l}}$. A line with Plücker coordinates ($\hat{\boldsymbol{l}}, \boldsymbol{m}$) can be represented by a DQ, as,

$$\check{\boldsymbol{l}} = \hat{\boldsymbol{l}} + \boldsymbol{m}\,\epsilon \tag{3.50}$$

If this line undergoes rigid transformation by rotation and then translation to get a line $\check{\boldsymbol{l}}'$, a DQ with the required rotation $\boldsymbol{q_r}$ and translation $\boldsymbol{t}$ can be used to achieve it. The DQ for transformation is given as $\check{\boldsymbol{q}} = \boldsymbol{q_r} + \boldsymbol{t}\boldsymbol{q_r}\,\frac{\epsilon}{2}$.

$$\check{\boldsymbol{l}}' = \check{\boldsymbol{q}} \otimes \check{\boldsymbol{l}} \otimes \check{\boldsymbol{q}}^* \tag{3.51}$$

It is to be noticed, that the operation involves the second conjugate as discussed in the subsection conjugates in the section of dual quaternions. It is not the third conjugate as used in the previous transformations in the previous sections due to the difference in the DQ forms of a point and a line (Jia (2013)). The operation for transformation yields the following result,

$$\check{\boldsymbol{l}}' = \boldsymbol{q_r}\hat{\boldsymbol{l}}\boldsymbol{q_r}^* + \left(\frac{1}{2}\boldsymbol{q_r}\hat{\boldsymbol{l}}\boldsymbol{q_r}^*\boldsymbol{t}^* + \boldsymbol{q_r}\boldsymbol{m}\boldsymbol{q_r}^* + \frac{1}{2}\boldsymbol{t}\boldsymbol{q_r}\hat{\boldsymbol{l}}\boldsymbol{q_r}^*\right)\epsilon \tag{3.52}$$

The above result can be verified easily and for the proof please refer to Jia (2013) in the references.

3.4.2. Screw Displacement

Mozzi-Chasles' theorem states, that any rigid displacement is equivalent to rotation about a certain line called the screw axis, followed or preceded by a translation along the same axis. The screw axis can be represented by the coordinates ($\hat{\boldsymbol{l}}$, $\boldsymbol{m}$). From Figure 3.2, the body is rotated by an angle θ and translated by the vector $\boldsymbol{t}$. The distance traversed by the body along the direction of the line $\hat{\boldsymbol{l}}$ is given by $d = \boldsymbol{t} \cdot \hat{\boldsymbol{l}}$. The screw motion is thus parameterised by $\hat{\boldsymbol{l}}$, $\boldsymbol{m}$, θ and d.

The transformation is performed by the DQ, $\check{\boldsymbol{q}} = \boldsymbol{q_r} + \boldsymbol{q_d}\,\epsilon$. As can be seen from the Figure 3.2, the rotation axis is the unit vector $\hat{\boldsymbol{l}}$ and the angle of rotation θ, hence

$$\boldsymbol{q_r} = \cos\frac{\theta}{2} + \hat{\boldsymbol{l}}\sin\frac{\theta}{2} \tag{3.53}$$

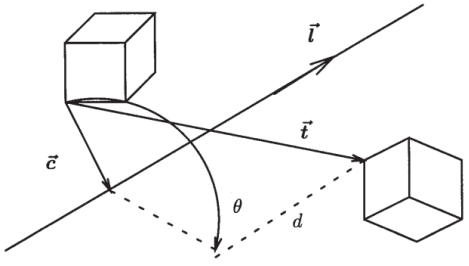


Figure 3.2: The geometry of a screw: every motion can be modeled as a rotation with angle θ about an axis at $\vec{\boldsymbol{c}}$, with direction $\hat{\boldsymbol{l}}$ and a subsequent translation d along the axis (Daniilidis (1999))

and

$$\boldsymbol{q_d} = \frac{1}{2}\boldsymbol{t}\boldsymbol{q_r} \tag{3.54}$$

The moment vector has been derived by Daniilidis (1999) and is of the resulting form:

$$\begin{aligned}\boldsymbol{m} &= \frac{1}{2}\left(\boldsymbol{t}\times\hat{\boldsymbol{l}} + \hat{\boldsymbol{l}}\times(\boldsymbol{t}\times\hat{\boldsymbol{l}})\cot\frac{\theta}{2}\right)\\ &= \frac{1}{2}\left(\boldsymbol{t}\times\hat{\boldsymbol{l}} + (\boldsymbol{t} - d\,\hat{\boldsymbol{l}})\cot\frac{\theta}{2}\right)\end{aligned} \tag{3.55}$$

Multiplying both sides of Eq. (3.55) with $\sin\frac{\theta}{2}$, we get:

$$\begin{aligned}\sin\frac{\theta}{2}\boldsymbol{m} &= \frac{1}{2}\left((\boldsymbol{t}\times\hat{\boldsymbol{l}})\sin\frac{\theta}{2} + (\boldsymbol{t} - d\hat{\boldsymbol{l}})\cos\frac{\theta}{2}\right)\\ &= \frac{1}{2}\boldsymbol{t}\cos\frac{\theta}{2} - \frac{d}{2}\hat{\boldsymbol{l}}\cos\frac{\theta}{2} + \frac{1}{2}(\boldsymbol{t}\times\hat{\boldsymbol{l}})\sin\frac{\theta}{2}\end{aligned} \tag{3.56}$$

Combining Eqs. (3.53), (3.54) and (3.55)

$$\begin{aligned}\frac{1}{2}\boldsymbol{t}\boldsymbol{q_r} &= \frac{1}{2}\boldsymbol{t}\left(\cos\frac{\theta}{2} + \hat{\boldsymbol{l}}\sin\frac{\theta}{2}\right)\\ &= \frac{1}{2}\boldsymbol{t}\cos\frac{\theta}{2} - \frac{1}{2}\boldsymbol{t}\cdot\hat{\boldsymbol{l}}\sin\frac{\theta}{2} + \frac{1}{2}\boldsymbol{t}\times\hat{\boldsymbol{l}}\sin\frac{\theta}{2}\\ &= -\frac{d}{2}\sin\frac{\theta}{2} + \sin\frac{\theta}{2}\boldsymbol{m} + \frac{d}{2}\hat{\boldsymbol{l}}\cos\frac{\theta}{2}\end{aligned} \tag{3.57}$$

The DQ for transformation now becomes:

$$\begin{aligned}\breve{\boldsymbol{q}} &= \left(\cos\frac{\theta}{2} + \hat{\boldsymbol{l}}\,sin\frac{\theta}{2}\right) + \left(-\frac{d}{2}\sin\frac{\theta}{2} + \sin\frac{\theta}{2}\boldsymbol{m} + \frac{d}{2}\hat{\boldsymbol{l}}\cos\frac{\theta}{2}\right)\epsilon\\ &= \left(\cos\frac{\theta}{2} - \frac{d\,\epsilon}{2}\sin\frac{\theta}{2}\right) + \left(\sin\frac{\theta}{2}(\hat{\boldsymbol{l}} + \boldsymbol{m}\,\epsilon) + \frac{d\,\epsilon}{2}\cos\frac{\theta}{2}(\hat{\boldsymbol{l}} + \boldsymbol{m}\,\epsilon)\right)\\ &= \left(\cos\frac{\theta}{2} - \frac{d}{2}\sin\frac{\theta}{2}\,\epsilon\right) + \left(\sin\frac{\theta}{2} + \frac{d}{2}\cos\frac{\theta}{2}\,\epsilon\right)(\hat{\boldsymbol{l}} + \boldsymbol{m}\,\epsilon)\end{aligned} \tag{3.58}$$

Using the Taylor series for expansion of trigonometric functions of dual numbers from Eq. (3.58),

$$\sin\frac{\theta + d\,\epsilon}{2} = \sin\frac{\theta}{2} + \frac{d}{2}\cos\frac{\theta}{2}\epsilon \tag{3.59a}$$

$$\cos\frac{\theta + d\,\epsilon}{2} = \cos\frac{\theta}{2} - \frac{d}{2}\sin\frac{\theta}{2}\epsilon \tag{3.59b}$$

The DQ using the above results becomes more compact:

$$\begin{aligned}\breve{\boldsymbol{q}} &= \cos\frac{\theta + d\,\epsilon}{2} + \sin\frac{\theta + d\,\epsilon}{2}(\hat{\boldsymbol{l}} + \boldsymbol{m}\,\epsilon)\\ \breve{\boldsymbol{q}} &= \cos\frac{\bar{\theta}}{2} + \bar{\hat{\boldsymbol{l}}}\,\sin\frac{\bar{\theta}}{2}\end{aligned} \tag{3.60}$$

where, $\bar{\theta}$ and $\bar{\hat{l}}$ are introduced as dual angle and dual vector. As can be seen this resembles the form of the rotation quaternion. The DQ $\breve{\boldsymbol{q}}$ is a unit DQ, which can be proved with conjugate multiplication as done before. This succinct form of DQ enables another property as a definition of powers ($\tau \geq 0$) of the DQ itself.

$$\breve{\boldsymbol{q}}^{\tau} = \cos\frac{\tau\bar{\theta}}{2} + \bar{\hat{\boldsymbol{l}}}\,\sin\frac{\tau\bar{\theta}}{2} \tag{3.61}$$

This means, that with re-multiplication of the line DQ with the same quaternion the rotation can be scaled as many times as the power of the transformation DQ, which is intuitive.

3.4.3. Screw Linear Interpolation

Spherical linear interpolation (SLERP) carried out by quaternions for smooth motion between two transforms has been extended to DQ operations as screw linear interpolation (ScLERP). It helps to achieve smooth interpolation between rotational and translational transforms. In SLERP, the quaternion for one transformation quaternion to another is achieved by (Kavan et al. (2007)):

$$SLERP\ (\tau;\ \boldsymbol{q_1},\ \boldsymbol{q_2}) = \boldsymbol{q_1} \otimes (\boldsymbol{q_1}^* \otimes \boldsymbol{q_2})^{\tau} \quad \tau \in [0,1] \tag{3.62}$$

Similarly in ScLERP,

$$ScLERP\ (\tau;\ \check{\boldsymbol{q}}_1,\ \check{\boldsymbol{q}}_2) = \check{\boldsymbol{q}}_1 \check{\otimes} (\check{\boldsymbol{q}}_1{}^* \check{\otimes} \check{\boldsymbol{q}}_2)^{\tau} \quad \tau \in [0,1] \tag{3.63}$$

The DQs are represented as:

$$\check{\boldsymbol{q}}_1 = \boldsymbol{q}_{r_1} + \frac{1}{2}\boldsymbol{t}_1 \boldsymbol{q}_{r_1}\,\epsilon \tag{3.64}$$

$$\check{\boldsymbol{q}}_2 = \boldsymbol{q}_{r_2} + \frac{1}{2}\boldsymbol{t}_2 \boldsymbol{q}_{r_2}\,\epsilon \tag{3.65}$$

The ScLERP operation results in:

$$\begin{aligned}
\check{\boldsymbol{q}}_1{}^* \otimes \check{\boldsymbol{q}}_2 &= \left(\boldsymbol{q}_{r_1}^* + \frac{1}{2}(\boldsymbol{t}_1 \boldsymbol{q}_{r_1})^*\,\epsilon\right) \otimes \left(\boldsymbol{q}_{r_2} + \frac{1}{2}\boldsymbol{t}_2 \boldsymbol{q}_{r_2}\,\epsilon\right) \\
&= \left(\boldsymbol{q}_{r_1}^* - \frac{1}{2}\boldsymbol{q}_{r_1}^* \boldsymbol{t}_1\,\epsilon\right) \check{\otimes} \left(\boldsymbol{q}_{r_2} + \frac{1}{2}\boldsymbol{t}_2 \boldsymbol{q}_{r_2}\,\epsilon\right) \\
&= \boldsymbol{q}_{r_1}^* \boldsymbol{q}_{r_2} + \frac{1}{2}\left(-\boldsymbol{q}_{r_1}^* \boldsymbol{t}_1 \boldsymbol{q}_{r_2} + \boldsymbol{q}_{r_1}^* \boldsymbol{t}_2 \boldsymbol{q}_{r_2}\right)\epsilon \\
&= \boldsymbol{q}_{r_1}^* \boldsymbol{q}_{r_2} + \frac{1}{2}\left(\boldsymbol{q}_{r_1}^* (\boldsymbol{t}_2 - \boldsymbol{t}_1) \boldsymbol{q}_{r_2}\right)\epsilon
\end{aligned} \tag{3.66}$$

As can be seen the new dual quaternion transforms the DQ from $\check{\boldsymbol{q}}_1$ to $\check{\boldsymbol{q}}_2$. Using the screw coordinates and Eq. (3.66) for higher powers of the DQ, we get the ScLERP operation as:

$$ScLERP\ (\tau;\ \check{\boldsymbol{q}}_1,\ \check{\boldsymbol{q}}_2) = \left(\boldsymbol{q}_{r_1} + \frac{1}{2}\boldsymbol{t}_1 \boldsymbol{q}_{r_1}\right) \otimes \left(\cos\frac{\tau\bar{\theta}}{2} + \bar{\hat{\boldsymbol{l}}}\,\sin\frac{\tau\bar{\theta}}{2}\right) \quad \tau \in [0,1] \tag{3.67}$$

The above result can be further expanded using the Eq. (3.58), to be written in terms of the screw coordinates $\hat{\boldsymbol{l}}$, $\boldsymbol{m}$, θ and d. ScLERP guarantees the shortest path and a constant speed interpolation. This can be seen in the above result, since the vector $\hat{\boldsymbol{l}}$ is independent of the τ times the half dual angle and therefore, the screw motion is constant. The angle of rotation as well as the translation both linearly depend on τ, the interpolation parameter. ScLERP is found to be independent of the choice of the coordinate system. For more details about the proof please refer to the work done by Kavan et al. (2007). Therefore, ScLERP is analogous to SLERP and does interpolation for rigid transformations whereas the latter does it for rotations.

3.5. Dual Quaternion Example

In this section we present with a simple example of rotating a vector using DQ cross product from Eq. (3.42). Consider the initial reference frame, A to be located at the origin and the other reference frame, B at $\boldsymbol{r}_{\mathtt{A}} = [4\ 3\ 2]^{\mathtt{T}}$ and to be rotated by 180° about the X-axis giving it the new attitude quaternion, $\boldsymbol{q}_{\mathtt{B/A}} = [1\ 0\ 0\ 0]^{\mathtt{T}}$. Let the vector, $\boldsymbol{v}_{\mathtt{A}} = [1\ 1\ 1]^{\mathtt{T}}$. We need to rotate and translate this vector to be represented in the B frame.

First, we need to compute the translational quaternion as below,

$$\boldsymbol{q}_d = \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}_A = \frac{1}{2}\begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}\begin{pmatrix} r_x \\ r_y \\ r_z \\ 0 \end{pmatrix};$$

$$= \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}\begin{pmatrix} 4 \\ 3 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 1.5 \\ -2 \end{pmatrix}$$

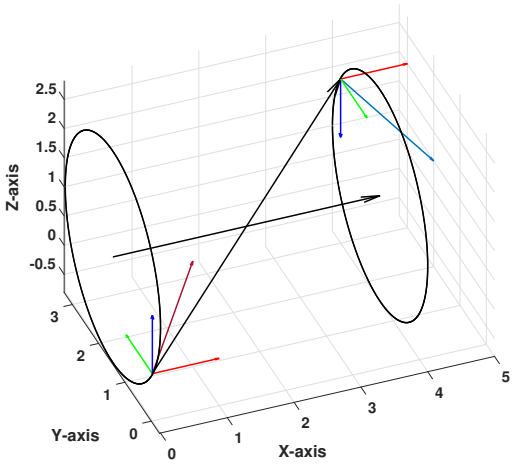


Figure 3.3: DQ rotation and translation example of a vector, with X, Y and Z-axes of object represented by red, green and blue arrows, respectively. The original vector is shown in dark red with its rotated and translated version shown in light blue.

The second step is to get the DQ and its conjugate,

$$\check{\boldsymbol{q}}_{\mathrm{B/A}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1.5 \\ -2 \end{pmatrix}, \qquad \check{\boldsymbol{q}}^{\diamond}_{\mathrm{B/A}} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1.5 \\ 2 \end{pmatrix}$$

The last step is to get the transformed vector, v_B, for which we use the operation,

$$1 + (\boldsymbol{v}_B + \boldsymbol{r}_A)\epsilon = \check{\boldsymbol{q}}_{\mathrm{B/A}} \otimes (1 + \boldsymbol{v}_A\epsilon) \otimes \check{\boldsymbol{q}}^{\diamond}_{\mathrm{B/A}} \tag{3.68}$$

We subtract the translation vector from the output of the to plot it and it is shown in Figure 3.3. The red vector is $\boldsymbol{v}_A$ and the initial reference frame is shown and the transformed vector, $\boldsymbol{v}_B$ is shown in blue, with the rotated co-ordinate frame.

4

Asteroid and Spacecraft Orbital and Attitude Dynamics

This chapter discusses the kinematics and dynamics involved with the motion of the asteroid and the SC. The dynamics of motion deal with the motion of the body caused by forces or moments acting on it. Kinematics provides with the position and attitude of the asteroid or SC with respect to a frame of reference, due to translational and angular velocities caused by the forces and torques acting on them. Gravitational forces due to the central body or a perturbing third body, radiative forces due to solar pulses or winds act on both the asteroid and SC. Propulsive forces due to the engine act on the SC. The SC is subjected to multiple perturbations due to the irregular gravity field of the small body, solar tides and SRP. In this chapter, we will discuss the kinematics and dynamics of motion; their mathematical representation for the asteroid and SC due to various forces acting on them and the required frames of reference. The kinematics and dynamics of the SC in terms of dual quaternions are discussed in this chapter, and they have been derived for a relative frame of reference as well.

4.1. Reference Frames

In order to represent the motions of the asteroid and spacecraft, appropriate frames of reference are required. In reality we do not have any "true" inertial frames, but depending on the system under consideration we can neglect accelerations and rotation of frames and consider them to be inertial (called pseudo-inertial). The reference frames to be used in this thesis are either pseudo-inertial or non-inertial and right-handed.

The SC position and attitude can be defined inertially as well as relatively with respect to the target body with the help of these frames. The reference frames help in carrying out operations to help orient the SC towards the target/Sun/Earth for mission objectives given in Section 2.3.2. The conversions from one frame of reference to another have also been discussed below. Figure 4.1 to 4.3 represent the different reference frames required for the mission.

Heliocentric frame - Ecliptic J2000 (H)
This frame of reference is defined by the ecliptic plane of the Earth. The X-axis of the frame is towards the vernal equinox, Z-axis along the angular momentum axis of the orbital plane of the Earth and Y-axis completes the right-handed set of the system. This frame is a pseudo inertial reference frame.

Asteroid Orbital frame at Pericentre(P)
This frame of reference is fixed at the pericentre of the target body's orbit around the Sun, where, the Z-axis of the frame coincides with the target body's axis of rotation, the X-axis along the nodal line of the orbital plane and Y-axis completes the right-handed set. Osculation of the orbit need not be considered for the time-frame of the mission at hand. It is, therefore, a pseudo inertial reference frame.

Asteroid Orbital frame (L)
This frame of reference is defined by the orbital plane of the target body. The Kepler elements of this orbit are given with respect to the Heliocentric reference frame. The X-axis of this frame is along the direction of motion of the asteroid, Z-axis is towards the zenith, i.e., always points towards the Sun and Y-axis completes

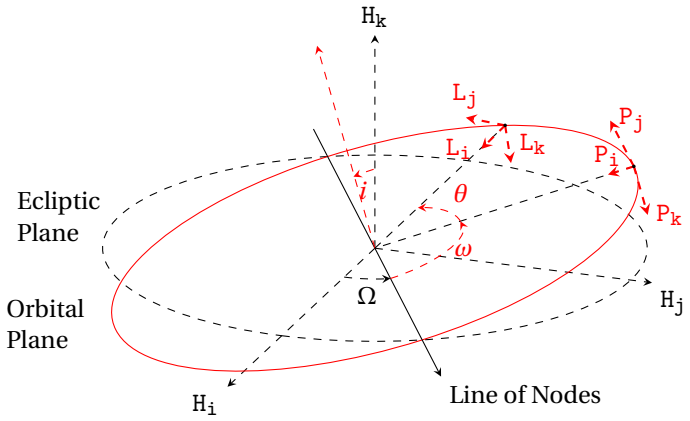


Figure 4.1: Representation of reference frames, H, P and L.

the right-handed set, such that it is along the negative angular momentum axis of the orbital plane. This is a rotating, non-inertial frame of reference.

Asteroid Inertial frame (I)
This frame of reference is fixed to the target body, with its centre at the target's centre of mass (com), axes fixed inertially, and Z-axis is coinciding with the target body's axis of rotation. It is a non-rotating and non-accelerating frame for the time-frame of the mission. Hence, it is a pseudo-inertial frame.

Asteroid Rotating frame (A)
This reference frame also has its centre at the target's com, but its axes are not inertially fixed and rotate along with the target body, such that they coincide with the principal moments of inertia of the target body. This is, therefore, a rotating and non-inertial frame of reference.

Body reference frame (B)
This reference frame is centred at the com of the SC, with axes along the principal moments of inertia of the SC. Since, the SC can change its attitude as required this frame of reference is not inertially fixed and therefore, is a non-inertial reference frame.

SC Orbital frame (S)
This frame is the local vertical local horizontal reference frame with its origin at the SC's COM. The Z-axis of this frame is directed towards the asteroid COM or nadir direction, while X-axis is along the direction of motion of the SC about the asteroid and Y-axis completes the right-handed triad, such that it is along the

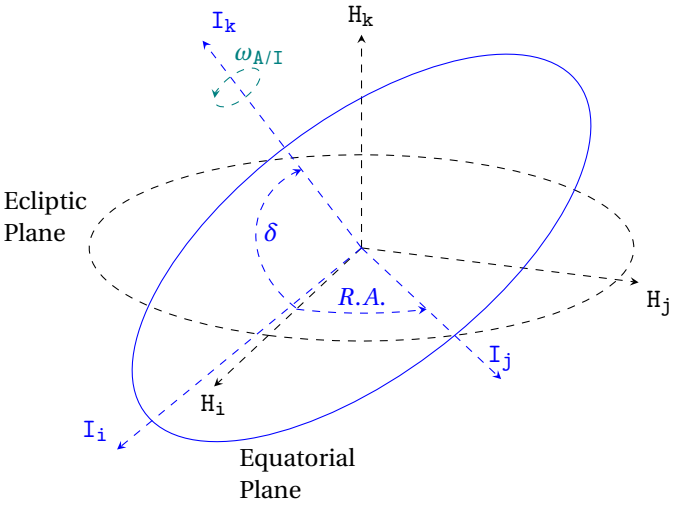


Figure 4.2: Representation of reference frames, H and I.

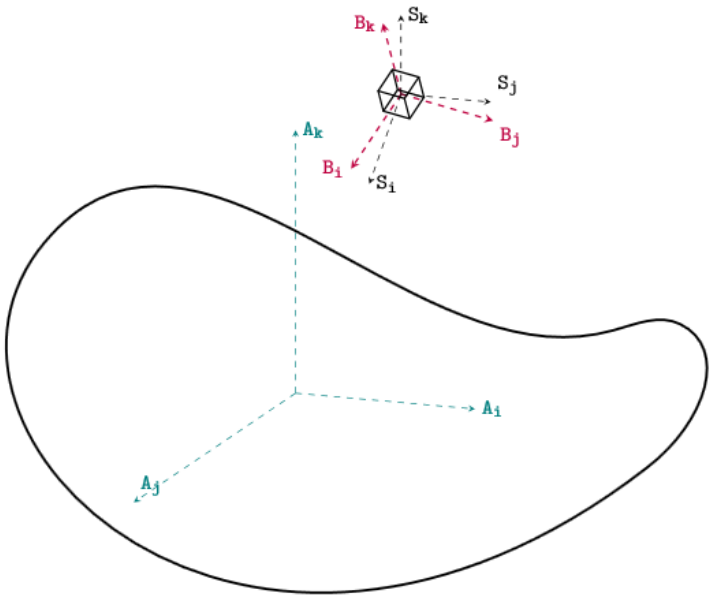


Figure 4.3: Representation of reference frames, A, B and S.

orbital angular momentum axis. This is also a non-inertial reference frame, since it is not inertially fixed in space.

4.2. Attitude Representation Methods

In this section, Euler's eigenaxis rotation and quaternions used for attitude representation are discussed, before getting into dual quaternion representation of SC attitude and its transformations. A detailed description of the common methods of DCMs, Euler angles and MRPs can be found in Markley and Crassidis (2013) and a brief discussion of the same can be found in Appendix C.

4.2.1. Euler's Eigenaxis Rotation and Quaternions

An Euler axis or eigenaxis is fixed to the body and is stationary with respect to an inertial frame, that is its orientation relative to the body frame and inertial frame remains unchanged throughout the motion of the SC. Rotating about such an axis an SC can change from its present attitude to any required attitude as shown in Figure 4.4. Considering the same two axes as before, body reference frame (B) and the inertial reference

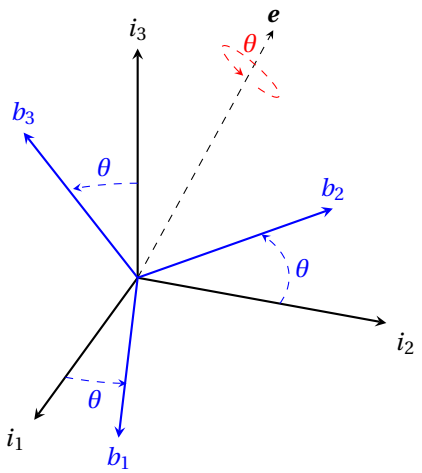


Figure 4.4: Euler's eigenaxis rotation visualised.

frame (I) with the Euler axis ($\boldsymbol{e}$),

$$\boldsymbol{e} = e_1\ \hat{i}_1 + e_2\ \hat{i}_2 + e_3\ \hat{i}_3 \tag{4.1}$$

$$= e_1\ \hat{b}_1 + e_2\ \hat{b}_2 + e_3\ \hat{b}_3 \tag{4.2}$$

where, $e_1,\ e_2,\ e_3$ are constrained by $e_1^2 + e_2^2 + e_3^2 = 1$. Rotating frame $\mathtt{I}$, such that $\hat{a}_1$ aligns with the direction of the axis **e**. Now the DCM for this conversion to a new frame $\mathtt{A}'$ can be given as,

$$\begin{pmatrix} \hat{i}_1' \\ \hat{i}_2' \\ \hat{i}_3' \end{pmatrix} = \boldsymbol{C}_{\mathtt{I}'/\mathtt{I}} \begin{pmatrix} \hat{i}_1 \\ \hat{i}_2 \\ \hat{i}_3 \end{pmatrix} = \begin{bmatrix} e_1 & e_2 & e_3 \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{pmatrix} \hat{i}_1 \\ \hat{i}_2 \\ \hat{i}_3 \end{pmatrix} \tag{4.3}$$

Now rotating both frames $\mathtt{I}$ and $\mathtt{I}'$ about the axis **e** by and angle θ, the frame $\mathtt{I}$ aligns with frame $\mathtt{B}$ whereas, frame $\mathtt{I}'$ forms a new frame $\mathtt{I}''$. In this situation the relative orientation of $\mathtt{I}''$ to $\mathtt{B}$ is the same as $\mathtt{I}'$ to $\mathtt{I}$. Therefore, $\boldsymbol{C}_{\mathtt{I}'/\mathtt{I}} = \boldsymbol{C}_{\mathtt{I}''/\mathtt{B}}$.

$$\boldsymbol{C}_{\mathtt{B}/\mathtt{I}} = \boldsymbol{C}_{\mathtt{B}/\mathtt{I}''}\ \boldsymbol{C}_{\mathtt{I}''/\mathtt{I}'}\ \boldsymbol{C}_{\mathtt{I}'/\mathtt{I}} = \boldsymbol{C}_{\mathtt{I}'/\mathtt{I}}^T\ \boldsymbol{C}_{\mathtt{I}''/\mathtt{I}'}\ \boldsymbol{C}_{\mathtt{I}'/\mathtt{I}} \tag{4.4}$$

Using DCMs as given below,

$$\boldsymbol{C}_{\mathtt{B}/\mathtt{I}} = \begin{bmatrix} \cos\theta + e_1^2(1-\cos\theta) & e_1e_2(1-\cos\theta) + e_3\ \sin\theta & e_1e_3(1-\cos\theta) - e_2\ \sin\theta \\ e_2e_1(1-\cos\theta) + e_3\ \sin\theta & \cos\theta + e_2^2(1-\cos\theta) & e_2e_3(1-\cos\theta) + e_1\ \sin\theta \\ e_3e_1(1-\cos\theta) + e_2\ \sin\theta & e_3e_2(1-\cos\theta) - e_1\ \sin\theta & \cos\theta + e_3^2(1-\cos\theta) \end{bmatrix} \tag{4.5}$$

The DCM can also be represented as,

$$\boldsymbol{C}_{\mathtt{B}/\mathtt{I}} = \boldsymbol{I}_3 \cos\theta + (1-\cos\theta)\boldsymbol{e}\boldsymbol{e}^T - [\boldsymbol{e}\times]\sin\theta \tag{4.6}$$

For $-180^\circ < \theta < 180^\circ$, using Eqs. (4.5), it can be found that,

$$\boldsymbol{e} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \frac{1}{2\sin\theta} \begin{pmatrix} C_{23} - C_{32} \\ C_{31} - C_{13} \\ C_{12} - C_{21} \end{pmatrix} \tag{4.7}$$

And the angle θ about the Euler axis $\boldsymbol{e}$ for the rotation of frame $\mathtt{A}$ to $\mathtt{B}$ can be found by,

$$\cos\theta = \frac{1}{2}(C_{11} + C_{22} + C_{33} - 1) = \frac{1}{2}(\mathrm{tr}C - 1) \tag{4.8}$$

Conversion of the frame obtained from above, that is $\mathtt{B}$ to a new frame, say $\mathtt{R}$, both with respect to $\mathtt{A}$ taken about two Euler axes $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$, respectively, can be done by,

$$\boldsymbol{C}_{\mathtt{B}/\mathtt{R}}(\boldsymbol{e},\theta) = \boldsymbol{C}_{\mathtt{B}/\mathtt{A}}(\boldsymbol{e}_1,\theta_2)\ \boldsymbol{C}'_{\mathtt{R}/\mathtt{A}}(\boldsymbol{e}_2,\theta_1) \tag{4.9}$$

The equivalent angle θ with the half angle theorem can be proved to be as below, γ being the angle between the two eigenaxes.

$$\cos\frac{\theta}{2} = \cos\frac{\theta_1}{2}\ \cos\frac{\theta_2}{2} - \sin\frac{\theta_1}{2}\ \sin\frac{\theta_2}{2}\ \cos\gamma \tag{4.10}$$

The initial introduction of quaternions by Sir William Rowan Hamilton in 1866 brought forward a new compact mathematical representation of concepts like rotation in three dimensions (McDonald (2010)). A quaternion represents two things, the three axes about which rotation occurs and the amount of rotation to be done. The general form in which quaternions are expressed is given as,

$$\boldsymbol{q} = q_1\ i + q_2\ j + q_3\ k + q_4 \tag{4.11}$$

where, q is a quaternion, $q_1, q_2, q_3, q_4 \in \mathbb{R}$ and $i,\ j$ and k are imaginary numbers having the following properties,

$$ij = k,\ jk = i,\ ki = j \tag{4.12}$$

$$ji = -k,\ kj = -i,\ ik = -j \tag{4.13}$$

and

$$i^2 = j^2 = k^2 = ijk = -1 \tag{4.14}$$

When Hamilton introduced quaternions, vector notation was yet to be introduced and hence the quaternions we use for attitude analysis are conceptually different. Quaternions can be represented as ordered pairs, as follows,

$$\boldsymbol{q} = \begin{pmatrix} \boldsymbol{q}_{1:3} \\ q_4 \end{pmatrix} \quad \text{where, } \boldsymbol{q}_{1:3} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{4.15}$$

where, $q_{1:3}$ represents the vector part and q_4 the scalar part. The product of two quaternions as introduced by Hamilton can be given as,

$$\boldsymbol{pq} = p_4 q_4 - \boldsymbol{p}_{1:3} \cdot \boldsymbol{q}_{1:3} + p_4 \boldsymbol{q}_{1:3} + q_4 \boldsymbol{p}_{1:3} + \boldsymbol{p}_{1:3} \times \boldsymbol{q}_{1:3} \tag{4.16}$$

The two most important operations with quaternions are a two different products of a pair of quaternions $\boldsymbol{p}$ and $\boldsymbol{q}$ (Markley and Crassidis (2013)),

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{pmatrix} q_4 \boldsymbol{p}_{1:3} + p_4 \boldsymbol{q}_{1:3} - \boldsymbol{p}_{1:3} \times \boldsymbol{q}_{1:3} \\ p_4 q_4 - \boldsymbol{p}_{1:3} \cdot \boldsymbol{q}_{1:3} \end{pmatrix} \tag{4.17}$$

$$\boldsymbol{p} \odot \boldsymbol{q} = \begin{pmatrix} q_4 \boldsymbol{p}_{1:3} + p_4 \boldsymbol{q}_{1:3} + \boldsymbol{p}_{1:3} \times \boldsymbol{q}_{1:3} \\ p_4 q_4 - \boldsymbol{p}_{1:3} \cdot \boldsymbol{q}_{1:3} \end{pmatrix} \tag{4.18}$$

From the above definitions we notice, that,

$$\boldsymbol{p} \otimes \boldsymbol{q} = \boldsymbol{q} \odot \boldsymbol{p} \tag{4.19}$$

The second product is equivalent to the one Hamilton introduced, but the first product is the most used for analysis of the attitude of the SC. It represents a similar composition of rotations as we obtain from a DCM. To achieve a quaternion from quaternions of two different frame conversions say $\boldsymbol{q}_{\mathrm{B/A}}$ and $\boldsymbol{q}_{\mathrm{A/R}}$, the product $\boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{q}_{\mathrm{A/R}} = \boldsymbol{q}_{\mathrm{A/R}} \odot \boldsymbol{q}_{\mathrm{B/A}}$ results in $\boldsymbol{q}_{\mathrm{B/R}}$. These products can be represented by matrix multiplication, as,

$$\boldsymbol{q} \otimes \bar{\boldsymbol{q}} = [\boldsymbol{q}\otimes]\bar{\boldsymbol{q}} = \bar{\boldsymbol{q}} \odot \boldsymbol{q} \tag{4.20}$$

$$\boldsymbol{q} \odot \bar{\boldsymbol{q}} = [\boldsymbol{q}\odot]\bar{\boldsymbol{q}} = \bar{\boldsymbol{q}} \otimes \boldsymbol{q} \tag{4.21}$$

where,

$$[\boldsymbol{q}\otimes] = \begin{bmatrix} q_4 \boldsymbol{I}_3 - [\boldsymbol{q}_{1:3}\times] & \boldsymbol{q}_{1:3} \\ -\boldsymbol{q}_{1:3}^T & q_4 \end{bmatrix} \tag{4.22}$$

$$[\boldsymbol{q}\odot] = \begin{bmatrix} q_4 \boldsymbol{I}_3 + [\boldsymbol{q}_{1:3}\times] & \boldsymbol{q}_{1:3} \\ -\boldsymbol{q}_{1:3}^T & q_4 \end{bmatrix} \tag{4.23}$$

Here, $[\boldsymbol{q}_{1:3}\times]$ denotes a cross product matrix. A general vector cross product matrix is shown as,

$$[\boldsymbol{q}_{1:3}\times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{4.24}$$

Quaternions are also known as Euler parameters. Considering the rotation of the Euler eigenaxis $\boldsymbol{n}$ about an arbitrary axis fixed in both the reference frames I and B by an angle θ, Euler parameters/quaternions can be defined as,

$$\boldsymbol{q}(\boldsymbol{n},\theta) = \begin{pmatrix} \sin\frac{\theta}{2}\boldsymbol{n} \\ \cos\frac{\theta}{2} \end{pmatrix} \tag{4.25}$$

Since, the rotation axis is represented by a unit vector, the quaternion defined by this should have a norm of one, i.e., $\|\boldsymbol{q}\| = 1$ an it is known as a unit quaternion. The Euler's parameters are constrained by the relation,

$$\boldsymbol{q}^{\boldsymbol{T}}\boldsymbol{q} + q_4^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{4.26}$$

Using the above results, the DCM can be parameterised with quaternions, as follows,

$$\begin{aligned} \boldsymbol{C}(\boldsymbol{q}, q_4) &= (q_4^2 - \|\boldsymbol{q}_{1:3}\|^2)\boldsymbol{I}_3 - 2q_4[\boldsymbol{q}_{1:3}\times] + 2\boldsymbol{q}_{1:3}\boldsymbol{q}_{1:3}^T \\ &= \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_2q_1-q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_3q_1+q_2q_4) & 2(q_3q_2-q_1q_4) & 1-2(q_1^2+q_2^2) \end{bmatrix} \end{aligned} \tag{4.27}$$

Quaternions are restricted by an Euler eigenaxis rotation angle range of $-180° \leq \theta \leq +180°$. This puts a constraint on continuous attitude tracking. However, quaternion representation of the kinematics are advantageous as compared to DCMs, since they involve lesser multiplications and to Rodrigues parameters and modified Rodrigues parameters by not resulting in bulky expressions (Razgus (2017)).

4.2.2. Dual Quaternions

As discussed in the previous section a dual quaternion can be used to transform a vector from one orientation to another. Here we will discuss how that is to be used for the representation of the position and attitude (pose) of an SC. As establish the DQ for transformation is given by, $\check{\boldsymbol{q}} = \boldsymbol{q}_r + \boldsymbol{q}_d\,\epsilon$. The real part of the DQ represents the attitude of the SC. For the position of the SC, the dual part has to include the position vector in terms of a pure quaternion from the frame of origin to the required frame of transformation.

The DQ has to be a unit quaternion, which means it has to have a norm of 1 to represent the pose. There are two constraints, that must be met by for the DQ to be a unit quaternion. The norm of the rotational quaternion should be 1 and the translation and rotational quaternions should be orthogonal to each other. Due to these two constraints, two DOFs are lost and we are left with six parameters out of eight, which is the number of parameters required to represent the pose of a SC.

$$\|\boldsymbol{q}_r\| = 1$$
$$\boldsymbol{q}_r \cdot \boldsymbol{q}_d = 0$$

The SC's orientation is defined with respect to a particular reference frame. If its attitude is represented in a frame B with respect to a frame A, it can be given by the unit quaternion $\boldsymbol{q}_{\mathtt{B/A}}$ and its position vector can be represented with respect to frame B by a pure quaternion $\boldsymbol{r}_{\mathtt{B}}$. Hence, the DQ for the SC orientation would come together as,

$$\check{\boldsymbol{q}}_{\mathtt{B/A}} = \boldsymbol{q}_{\mathtt{B/A}} + \frac{1}{2}\boldsymbol{r}_{\mathtt{B}} \otimes \boldsymbol{q}_{\mathtt{B/A}}\,\epsilon \tag{4.28}$$

To use the position vector with respect to the frame A, we simply need to use the operation for the transformation of quaternions,

$$\boldsymbol{r}_{\mathtt{A}} = \boldsymbol{q}_{\mathtt{A/B}} \otimes \boldsymbol{r}_{\mathtt{B}} \otimes \boldsymbol{q}^*_{\mathtt{A/B}} \tag{4.29}$$

Since, the rotation quaternions are unit in nature, to get the product for transforming the translation quaternion we multiply the dual part by $\boldsymbol{q}^*_{\mathtt{A/B}} \otimes \boldsymbol{q}_{\mathtt{A/B}} = I_4$, since they are unit quaternions of rotation. Also $\boldsymbol{q}_{\mathtt{B/A}}$ can be written as $\boldsymbol{q}^*_{\mathtt{A/B}}$. Hence we can write Eq. (4.28), as,

$$\begin{aligned}
\check{\boldsymbol{q}}_{\mathtt{B/A}} &= \boldsymbol{q}_{\mathtt{B/A}} + \frac{1}{2}\boldsymbol{q}^*_{\mathtt{A/B}} \otimes \boldsymbol{q}_{\mathtt{A/B}} \otimes \boldsymbol{r}_{\mathtt{B}} \otimes \boldsymbol{q}_{\mathtt{B/A}}\,\epsilon \\
&= \boldsymbol{q}_{\mathtt{B/A}} + \frac{1}{2}\boldsymbol{q}^*_{\mathtt{A/B}} \otimes \boldsymbol{q}_{\mathtt{A/B}} \otimes \boldsymbol{r}_{\mathtt{B}} \otimes \boldsymbol{q}^*_{\mathtt{A/B}}\,\epsilon \\
&= \boldsymbol{q}_{\mathtt{B/A}} + \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}_{\mathtt{A}}\,\epsilon
\end{aligned} \tag{4.30}$$

This shows the reversibility of multiplicands in the dual part gives the same DQ with a change in the reference frame of the translation vector. Therefore, from any DQ, the rotation and position quaternion can be obtained as,

$$\boldsymbol{q}_{\mathtt{B/A}} = \boldsymbol{q}_r \tag{4.31}$$
$$\boldsymbol{r}_{\mathtt{B}} = 2\boldsymbol{q}_d \otimes \boldsymbol{q}^*_r \tag{4.32}$$
$$\boldsymbol{r}_{\mathtt{A}} = 2\boldsymbol{q}^*_r \otimes \boldsymbol{q}_d \tag{4.33}$$

4.3. Reference Frame Transformations

In this section, we will discuss the conversion of reference frames from one form to another as required during various phases of the mission.

Heliocentric - Asteroid Orbital (H − L)
Initially, we convert the H-frame to the asteroid centred frame at the perigee of the orbit P and then this frame is rotated about the $\mathtt{P}_z$ axis by the true anomaly of the asteroid position, in orbit. This gives us the asteroid orbital L-frame. To convert from the H-frame to P-frame, we need a ZXZ rotation sequence with the Euler angles of the argument of perigee, the inclination of the orbit to the ecliptic and the right ascension of ascending node, i.e., $[\omega + \frac{\pi}{2},\ i,\ \Omega]$. Hence a position vector in the P-frame can be given with the help of the DCM matrix $\boldsymbol{C}_{\mathtt{P/H}}$ as,

$$\boldsymbol{r}_{\mathtt{P}} = \boldsymbol{C}_{\mathtt{P/H}}\,\boldsymbol{r}_{\mathtt{H}} \tag{4.34}$$

Let cos = c, and sin = s, the DCM $\boldsymbol{C}_{\mathtt{P/H}}$ can be given as,

$$\begin{pmatrix} \mathtt{P}_i \\ \mathtt{P}_j \\ \mathtt{P}_k \end{pmatrix} = \begin{bmatrix} c(\omega+\frac{\pi}{2})c(\Omega) - s(\omega+\frac{\pi}{2})s(\Omega)c(i) & c(\omega+\frac{\pi}{2})s(\Omega) + s(\omega+\frac{\pi}{2})c(\Omega)s(i) & s(\omega+\frac{\pi}{2})s(i) \\ -s(\omega+\frac{\pi}{2})c(\Omega) - c(\omega+\frac{\pi}{2})s(\Omega)c(i) & -s(\omega+\frac{\pi}{2})s(\Omega) + c(\omega+\frac{\pi}{2})c(\Omega)c(i) & c(\omega+\frac{\pi}{2})s(i) \\ s(\Omega)s(i) & -c(\Omega)s(i) & c(i) \end{bmatrix} \begin{pmatrix} \mathtt{H}_i \\ \mathtt{H}_j \\ \mathtt{H}_k \end{pmatrix} \tag{4.35}$$

The DCM can be converted to the rotation quaternion $\boldsymbol{q}_{\mathtt{P/H}}$ with the help of the Eq. (4.35). and the frame conversion from the P-frame to the L-frame,

$$\boldsymbol{q}_{\mathtt{L/H}} = \boldsymbol{q}_{\mathtt{L/P}} \otimes \boldsymbol{q}_{\mathtt{P/H}} = \boldsymbol{q}(\mathtt{P}_k, \theta) \otimes \boldsymbol{q}_{\mathtt{P/H}} \tag{4.36}$$

where, θ is the true anomaly of the asteroid and therefore, this quaternion needs to be updated with time as the asteroid orbits the Sun.

Heliocentric - Asteroid Inertial ($\mathtt{H-I}$)
The orientation of the asteroid rotation axis or in other words its axial tilt is given by the right ascension, λ and declination, β from the H-frame. Hence it can be directly transformed from the heliocentric frame. Since, this is a non-rotating inertial frame, it can be obtained directly from the P-frame. A sequence of rotations, first about the k-axis of the P-frame by the right ascension and then about the i-axis of the transformed frame by the

$$\boldsymbol{q}_{\mathtt{I/H}} = \boldsymbol{q}(\mathtt{I}_i, \beta) \otimes \boldsymbol{q}(\mathtt{H}_k, \lambda) \tag{4.37}$$

Asteroid Inertial - Asteroid Rotating ($\mathtt{I-A}$)
Transforming from the inertial frame to the rotating frame is simply the rotating the inertial frame about its K-axis at the rotation rate of the asteroid.

$$\boldsymbol{q}_{\mathtt{A/I}} = \boldsymbol{q}(\mathtt{I}_k, (\boldsymbol{\omega}_{\mathtt{A/I}}\, t)) \tag{4.38}$$

Heliocentric - Asteroid Rotating ($\mathtt{H-A}$)
This transformation can simply be obtained from the previous two transformations, as follows,

$$\boldsymbol{q}_{\mathtt{A/H}} = \boldsymbol{q}(\mathtt{I}_k, (\boldsymbol{\omega}_{\mathtt{A/I}}\, t)) \otimes \boldsymbol{q}(\mathtt{I}_i, \beta) \otimes \boldsymbol{q}(\mathtt{H}_k, \lambda) \tag{4.39}$$

Asteroid Inertial - SC orbital ($\mathtt{I-S}$)
Similar to the transformation from heliocentric to asteroid orbital frame the transformation from I-frame to S-frame can be carried out with the Euler angles $[\omega_{\mathtt{S}} + \frac{\pi}{2},\ (i_{\mathtt{S}}),\ (\Omega_{\mathtt{S}})]$. Let $\omega_{\mathtt{S}_1} = \omega_{\mathtt{S}} + \frac{\pi}{2}$,

$$\begin{pmatrix} \mathtt{S_P}_i \\ \mathtt{S_P}_j \\ \mathtt{S_P}_k \end{pmatrix} = \begin{bmatrix} c(\omega_{\mathtt{S}_1})c(\Omega_{\mathtt{S}}) - s(\omega_{\mathtt{S}_1})s(\Omega_{\mathtt{S}})c(i_{\mathtt{S}}) & c(\omega_{\mathtt{S}_1})s(\Omega_{\mathtt{S}}) + s(\omega_{\mathtt{S}_1})c(\Omega_{\mathtt{S}})s(i_{\mathtt{S}}) & s(\omega_{\mathtt{S}_1})s(i_{\mathtt{S}}) \\ -s(\omega_{\mathtt{S}_1})c(\Omega_{\mathtt{S}}) - c(\omega_{\mathtt{S}_1})s(\Omega_{\mathtt{S}})c(i_{\mathtt{S}}) & -s(\omega_{\mathtt{S}_1})s(\Omega_{\mathtt{S}}) + c(\omega_{\mathtt{S}_1})c(\Omega_{\mathtt{S}})c(i_{\mathtt{S}}) & c(\omega_{\mathtt{S}_1})s(i_{\mathtt{S}}) \\ s(\Omega_{\mathtt{S}})s(i_{\mathtt{S}}) & -c(\Omega_{\mathtt{S}})s(i_{\mathtt{S}}) & c(i_{\mathtt{S}}) \end{bmatrix} \begin{pmatrix} \mathtt{I}_i \\ \mathtt{I}_j \\ \mathtt{I}_k \end{pmatrix} \tag{4.40}$$

The DCM can be converted to the rotation quaternion $\boldsymbol{q}_{\mathtt{P/H}}$ with the help of the Eq. (4.27) and the frame conversion from the $\mathtt{S_P}$-frame to the S-frame,

$$\boldsymbol{q}_{\mathtt{S/I}} = \boldsymbol{q}_{\mathtt{S/S_P}} \otimes \boldsymbol{q}_{\mathtt{S_P/I}} = \boldsymbol{q}(\mathtt{S_P}_k, \theta_S) \otimes \boldsymbol{q}_{\mathtt{S_P/I}} \tag{4.41}$$

where, θ_S is the true anomaly of the SC.

Asteroid Rotating - SC orbital ($\mathtt{A-S}$)
The S - frame can be obtained from the A - frame as from Eqs. (4.38) and (4.41), as,

$$\boldsymbol{q}_{\mathtt{S/A}} = \boldsymbol{q}_{\mathtt{S/I}} \otimes \boldsymbol{q}_{\mathtt{I/A}} = \boldsymbol{q}_{\mathtt{S/I}} \otimes \boldsymbol{q}^*_{\mathtt{A/I}} = \boldsymbol{q}_{\mathtt{S/I}} \otimes \boldsymbol{q}^*(\mathtt{I}_k, (\boldsymbol{\omega}_{\mathtt{A/I}}\, t)) \tag{4.42}$$

SC orbital - Body reference ($\mathtt{S-B}$)
The orientation of the body reference frame of the SC is calculated with the help of the navigation system and then it can be obtained with respect to the SC orbital reference frame. The attitude controller is used to achieve the required orientation of the SC (B - frame) as per the mission objectives. The transformation can be obtained with the quaternion given in Eqs. (4.41) to get the quaternion, $\boldsymbol{q}_{\mathtt{B/S}}$, as,

$$\boldsymbol{q}_{\mathtt{B/S}} = \boldsymbol{q}_{\mathtt{B/I}} \otimes \boldsymbol{q}_{\mathtt{I/S}} = \boldsymbol{q}(\mathtt{I}_k, (\boldsymbol{\omega}_{\mathtt{B/I}}\, t)) \otimes \boldsymbol{q}^*_{\mathtt{S/I}} \tag{4.43}$$

Asteroid Rotating - Body reference ($\mathtt{A-B}$)
Using the quaternions $\boldsymbol{q}_{\mathtt{S/A}}$ and $\boldsymbol{q}_{\mathtt{B/S}}$ given in Eqs. (4.42) and (4.43) the required quaternion $\boldsymbol{q}_{\mathtt{B/A}}$ can be obtained for operating the SC as per the mission objectives and requirements.

$$\boldsymbol{q}_{\mathtt{B/A}} = \boldsymbol{q}_{\mathtt{B/S}} \otimes \boldsymbol{q}_{\mathtt{S/A}} = \boldsymbol{q}(\mathtt{I}_k, (\boldsymbol{\omega}_{\mathtt{B/I}}\, t)) \otimes \boldsymbol{q}^*(\mathtt{I}_k, (\boldsymbol{\omega}_{\mathtt{A/I}}\, t)) \tag{4.44}$$

4.4. State Vectors

In the thesis, we work with three state vectors: one that is represented with respect to the asteroid centred inertial frame (I), the second and third with respect to the asteroid centred rotating frame (A). They can be given as follows,

$$\boldsymbol{X}_{\mathtt{I}} = \begin{pmatrix} m \\ \boldsymbol{r}^{\mathtt{I}} \\ \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}} \\ \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}} \\ \boldsymbol{q}_{\mathtt{B/I}} \\ \boldsymbol{q}_{\mathtt{A/I}} \end{pmatrix}, \quad \boldsymbol{X}_{\mathtt{A}} = \begin{pmatrix} m \\ \boldsymbol{r}^{\mathtt{A}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} \\ \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \\ \boldsymbol{q}_{\mathtt{B/A}} \end{pmatrix}, \quad \check{\boldsymbol{X}}_{\mathtt{A}} = \begin{pmatrix} m \\ \check{\boldsymbol{q}}_{\mathtt{B/A}} \\ \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} \end{pmatrix} \tag{4.45}$$

where, m is the total mass of the SC at any instant of time. The dynamics using the inertial state vector will avoid additional accelerations due to the non-inertial asteroid rotating frame. The guidance algorithm will be developed for the inertial state first, to establish results comparable to Szmuk et al. (2017). The relative state vector will be used to verify kinematics and dynamics derived in the relative asteroid rotating frame, A. Since these 6DOF EOMs in the relative frame are used to develop the third state, i.e., relative dual quaternion state, the results from the algorithm using the dual quaternion state can be verified with the Cartesian-quaternion relative state.

Hence, in the next sections, we present the inertial frame kinematics and dynamics for translational and rotational motion. Then we derive them for the relative frame. Finally, the novel representation of EOMS with dual quaternions in a relative frame have been derived and presented in the most lucrative form possible.

4.5. Translational Motion (Orbital Motion)

Once the SC has been transferred to its operational orbit, it must be maintained in orbit with certain accuracy to carry out the different scientific objectives. The maintaining of the orbit needs to be done at minimum or acceptable expenditure of propellant. Assuming the SC has an appropriate mass, that does not affect the motion of the small body it orbits, we can establish its equation of motion.

Due to the presence of perturbing forces, the orbital motion cannot be characterised as Keplerian and is an evolving orbit called an osculating orbit. The equation of the orbital motion of the SC about the small body in the inertial frame fixed to the COM of the small body can, therefore, be given as (Sidi (1997)),

$$\ddot{\boldsymbol{r}}^{\mathtt{I}} = \boldsymbol{a}^{\mathtt{I}}_{G} + \boldsymbol{a}^{\mathtt{I}}_{D} \tag{4.46}$$

where, as established in the earlier chapters, $\boldsymbol{a}^{\mathtt{I}}_{G}$ is the acceleration due to gravity and $\boldsymbol{a}^{\mathtt{I}}_{D}$ represents the perturbing forces caused by the 3$^{\text{rd}}$ body perturbation of the Sun and the solar radiation pressure in the inertial frame of reference. The perturbing accelerations are appreciably smaller than the gravitational acceleration. Newton's laws hold true for inertial frames and therefore, we need to define the equations of motions based on them in inertial frames. We will establish all the equations of motion involved, in the inertial frame and in the next sections convert them to a relative frame.

4.5.1. Kinematics

The kinematic equation for the translational motion of the SC in the inertial frame can be written as,

$$\dot{\boldsymbol{r}}^{\mathtt{I}} = \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}} \tag{4.47}$$

where, $\boldsymbol{r}^{\mathtt{I}}$ and $\boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}}$ are the position and velocity of the SC with respect to the inertial frame, respectively.

4.5.2. Dynamics

The dynamic motion of the SC can be given by,

$$m\dot{\boldsymbol{v}}^{\mathtt{I}}_{\mathtt{B/I}} = \boldsymbol{F}^{\mathtt{I}} \tag{4.48}$$

$$\dot{\boldsymbol{v}}^{\mathtt{I}}_{\mathtt{B/I}} = \boldsymbol{a}^{\mathtt{I}}_{G} + \boldsymbol{a}^{\mathtt{I}}_{D} \tag{4.49}$$

where, m and $\boldsymbol{F}^{\mathtt{I}}$ are the mass of the SC and total force acting on the SC with respect to the inertial frame, respectively.

4.6. Rotational Motion

In this section, we will discuss the concepts of angular velocity or angular rate vector and present the kinematics and dynamics of the rotational motion of the SC about itself or the body reference frame.

4.6.1. Attitude Kinematics

The angular velocity of the body frame with respect to the inertial frame in the body frame ($\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}$) can be given by the equation below. Let ω_1, ω_2, ω_3 be the components of the angular velocity of the spacecraft in the body reference frame represented by the unit axes $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ and $\boldsymbol{b}_3$.

$$\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}} = \omega_1 \boldsymbol{b_1} + \omega_2 \boldsymbol{b_2} + \omega_3 \boldsymbol{b_3} = \begin{bmatrix} \boldsymbol{b_1} & \boldsymbol{b_2} & \boldsymbol{b_3} \end{bmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \tag{4.50}$$

The kinematic differential equation in terms of DCMs can be represented as (Markley and Crassidis (2013)),

$$\dot{\boldsymbol{C}}_{\mathtt{B/I}} = -[\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}} \times] \boldsymbol{C}_{\mathtt{B/I}} \tag{4.51}$$

where, $\boldsymbol{C}_{\mathtt{B/I}}$ is the attitude matrix of the body frame with respect to the inertial frame of reference. We have described the DCM in Eq. (4.27), using quaternions. Rewriting the kinematic differential equation using quaternions we get (Markley and Crassidis (2013)),

$$\dot{\boldsymbol{q}}_{\mathtt{B/I}} = \frac{1}{2} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}} \otimes \boldsymbol{q}_{\mathtt{B/I}} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}) \boldsymbol{q}_{\mathtt{B/I}} = \frac{1}{2} \Xi(\boldsymbol{q}_{\mathtt{B/I}}) \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}} \tag{4.52}$$

where, $\boldsymbol{\Omega}$ and Ξ are expressed as,

$$\boldsymbol{\Omega} = \begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{4.53}$$

$$\boldsymbol{\Xi} = \begin{bmatrix} q_4 \boldsymbol{I}_3 + [\boldsymbol{q}_{1:3}\times] \\ -\boldsymbol{q}^T_{1:3} \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{4.54}$$

The representation of the kinematics with Ξ is more convenient to use and depending on the requirement while developing the dual quaternion EOMs, these representations have been used interchangeably. As can be seen the kinematic equation with quaternions is in a linear relationship with the quaternion itself and does not have any trigonometric functions. Hence, it provides a less cumbersome and more appealing way of attitude representation.

4.6.2. Attitude Dynamics

The rotational equation of a rigid body about the origin of the inertial frame can be given as,

$$\int \boldsymbol{r}^{\mathtt{B}} \times \ddot{\boldsymbol{r}}^{\mathtt{I}} \mathrm{d}m = \boldsymbol{T}^{\mathtt{B}} \tag{4.55}$$

where, $\boldsymbol{r}^{\mathtt{B}}$ and $\boldsymbol{r}^{\mathtt{I}}$ are the position vectors of the SC with respect to the body frame and the inertial frame of reference, respectively, m is the mass of the SC and $\boldsymbol{T}^{\mathtt{B}}$ is the sum of all torques acting on the SC: disturbance and control torques. This equation can be further resolved and be written as (Wie (2001)),

$$\dot{\boldsymbol{H}}^{\mathtt{B}} + \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}} \times \boldsymbol{H}^{\mathtt{B}} = \boldsymbol{T}^{\mathtt{B}} \tag{4.56}$$

where, $\boldsymbol{H}^{\mathtt{B}}$ is the absolute angular momentum of the body about the origin of the body reference frame and $\boldsymbol{r}^{\mathtt{B}}_c$ is the position vector of the COM of the SC with respect to the body reference frame. The inertia matrix of the SC about the body-fixed reference frame can be given as,

$$\boldsymbol{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \tag{4.57}$$

Substituting $\boldsymbol{H}_{\mathrm{B}} = \boldsymbol{J}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}$ in Eq. (4.56), the dynamics of the SC can be given as,

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}} + \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} = \boldsymbol{T}^{\mathrm{B}} = \boldsymbol{T}^{\mathrm{B}}_{C} + \boldsymbol{T}^{\mathrm{B}}_{D} \tag{4.58}$$

where, $\boldsymbol{T}^{\mathrm{B}}_{C}$ and $\boldsymbol{T}^{\mathrm{B}}_{D}$ are the control and disturbance torques, respectively.

4.7. Asteroid Dynamics and Kinematics

Since, the simulation time is within the range of 100s of seconds for descent and not more than two days for mapping, the orbit of the asteroid around the Sun need not be simulated. For such a short duration, this movement can be assumed to be negligible and hence, we do not incorporate the target body orbital dynamics for the purpose of this thesis. But, as seen in Table 2.1, the selected asteroids have very short rotation periods and ignoring the asteroid rotation could prove to be dangerous for the SC proximity operations. Hence, the angular velocity of the asteroid is considered, but, will be assumed as a constant in both direction and magnitude, without the presence of precession and nutation of the rotation axis. The angular velocity of the asteroid with respect to the inertial frame of reference can be given as,

$$\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} = \text{constant} \tag{4.59}$$

Kinematics of the asteroid can be given in a similar fashion as the SC attitude kinematics,

$$\dot{\boldsymbol{q}}_{\mathrm{A/I}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}})\boldsymbol{q}_{\mathrm{A/I}} = \frac{1}{2}\Xi(\boldsymbol{q}_{\mathrm{A/I}})\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \tag{4.60}$$

4.8. Relative Dynamics and Kinematics

The kinematics and dynamics discussed above are all in the inertial frame of reference. Since state vectors in the relative frame need to be incorporated, the EOMs need to be defined with respect to the asteroid rotating frame. The relative position can be given by,

$$\boldsymbol{r}^{\mathrm{A}} = \boldsymbol{C}_{\mathrm{A/I}}\boldsymbol{r}^{\mathrm{I}} \tag{4.61}$$

where, $\boldsymbol{C}_{\mathrm{A/I}}$ can be expressed using the quaternion $\boldsymbol{q}_{\mathrm{A/I}}$ as

$$\begin{aligned}\boldsymbol{C}(\boldsymbol{q}, q_4) &= (q_4^2 - \|\boldsymbol{q}_{1:3}\|^2)\mathbf{I}_3 - 2q_4[\boldsymbol{q}_{1:3}\times] + 2\boldsymbol{q}_{1:3}\boldsymbol{q}_{1:3}^T \\ &= \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_2q_1-q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_3q_1+q_2q_4) & 2(q_3q_2-q_1q_4) & 1-2(q_1^2+q_3^2) \end{bmatrix}\end{aligned} \tag{4.62}$$

4.8.1. Translational Motion

The relative translational kinematic equation can be obtained by taking the derivative of the relative position vector, $\boldsymbol{r}^{\mathrm{A}}$,

$$\begin{aligned}\dot{\boldsymbol{r}}^{\mathrm{A}} &= \boldsymbol{C}_{\mathrm{A/I}}\dot{\boldsymbol{r}}^{\mathrm{I}} - \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{C}_{\mathrm{A/I}}\boldsymbol{r}^{\mathrm{I}} \\ \boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}} &= \boldsymbol{C}_{\mathrm{A/I}}\dot{\boldsymbol{r}}^{\mathrm{I}} - \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{r}^{\mathrm{A}}\end{aligned} \tag{4.63}$$

The relative translational dynamic equation can be obtained by taking the derivative of the relative velocity vector, $\boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}}$,

$$\dot{\boldsymbol{v}}^{\mathrm{A}}_{\mathrm{B/A}} = \boldsymbol{C}_{\mathrm{A/I}}\ddot{\boldsymbol{r}}^{\mathrm{I}} - \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{C}_{\mathrm{A/I}}\dot{\boldsymbol{r}}^{\mathrm{I}} - \dot{\boldsymbol{\omega}}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{C}_{\mathrm{A/I}}\boldsymbol{r}^{\mathrm{I}} + \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{C}_{\mathrm{A/I}}\boldsymbol{r}^{\mathrm{I}} - \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{C}_{\mathrm{A/I}}\dot{\boldsymbol{r}}^{\mathrm{I}} \tag{4.64}$$

The term with $\dot{\boldsymbol{\omega}}^{\mathrm{A}}_{\mathrm{A/I}}$ is omitted, since the asteroid rotation rate is assumed to be constant causing the angular acceleration to be zero. Now replacing $\boldsymbol{C}_{\mathrm{A/I}}\boldsymbol{r}^{\mathrm{I}}$ and $\boldsymbol{C}_{\mathrm{A/I}}\dot{\boldsymbol{r}}^{\mathrm{I}}$ from Eqs. (4.61) and (4.63), we get

$$\boldsymbol{a}^{\mathrm{A}}_{\mathrm{B/A}} = \boldsymbol{C}_{\mathrm{A/I}}\ddot{\boldsymbol{r}}^{I} - 2\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times (\boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}} + \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{r}^{\mathrm{A}}) + \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{r}^{\mathrm{A}} \tag{4.65}$$

The relative translational dynamic equation can therefore, be given as,

$$\boldsymbol{a}^{\mathrm{A}}_{\mathrm{B/A}} = \boldsymbol{C}_{\mathrm{A/I}}\ddot{\boldsymbol{r}}^{\mathrm{I}} - 2\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}} - \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{r}^{\mathrm{A}} \tag{4.66}$$

4.8.2. Rotational Motion

The relative attitude quaternion can be given as,

$$\boldsymbol{q}_{\mathrm{B/A}} = \boldsymbol{q}_{\mathrm{B/I}} \otimes \boldsymbol{q}^{*}_{\mathrm{A/I}} \tag{4.67}$$

and the relative angular rate,

$$\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} = \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} - \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \tag{4.68}$$

where, $\boldsymbol{C}_{\mathrm{B/A}}$ can be obtained from $\boldsymbol{q}_{\mathrm{B/A}}$. For the attitude kinematics, the derivative of the relative quaternion, $\boldsymbol{q}_{\mathrm{B/A}}$ is taken with the help of quaternion derivative properties (Jia (2018)),

$$\begin{aligned}
\dot{\boldsymbol{q}}_{\mathrm{B/A}} &= \dot{\boldsymbol{q}}_{\mathrm{B/I}} \otimes \boldsymbol{q}^{*}_{\mathrm{A/I}} + \boldsymbol{q}_{\mathrm{B/I}} \otimes \dot{\boldsymbol{q}}^{*}_{\mathrm{A/I}} \\
\dot{\boldsymbol{q}}_{\mathrm{B/A}} &= \frac{1}{2}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \otimes \boldsymbol{q}_{\mathrm{B/I}} \otimes \boldsymbol{q}^{*}_{\mathrm{A/I}} + \boldsymbol{q}_{\mathrm{B/I}} \otimes \frac{1}{2}\boldsymbol{\omega}^{\mathrm{I}}_{\mathrm{I/A}} \otimes \boldsymbol{q}_{\mathrm{I/A}}
\end{aligned} \tag{4.69}$$

From properties of angular velocities we know, that,

$$\boldsymbol{\omega}^{\mathrm{I}}_{\mathrm{I/A}} = -\boldsymbol{\omega}^{\mathrm{I}}_{\mathrm{A/I}} = -\boldsymbol{q}_{\mathrm{I/A}} \otimes \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \otimes \boldsymbol{q}^{*}_{\mathrm{I/A}} \tag{4.70}$$

Substituting this new representation in Eq. (4.69),

$$\begin{aligned}
\dot{\boldsymbol{q}}_{\mathrm{B/A}} &= \frac{1}{2}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \otimes \boldsymbol{q}_{\mathrm{B/A}} - \frac{1}{2}\boldsymbol{q}_{\mathrm{B/I}} \otimes \boldsymbol{q}_{\mathrm{I/A}} \otimes \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \otimes \boldsymbol{q}^{*}_{\mathrm{I/A}} \otimes \boldsymbol{q}_{\mathrm{I/A}} \\
\dot{\boldsymbol{q}}_{\mathrm{B/A}} &= \frac{1}{2}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \otimes \boldsymbol{q}_{\mathrm{B/A}} - \frac{1}{2}\boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}}
\end{aligned} \tag{4.71}$$

From Eq. (4.68) we get,

$$\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} = \boldsymbol{q}^{*}_{\mathrm{B/A}} \otimes (\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} - \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}) \otimes \boldsymbol{q}_{\mathrm{B/A}} \tag{4.72}$$

Using the above equation in Eq. (4.71),

$$\dot{\boldsymbol{q}}_{\mathrm{B/A}} = \frac{1}{2}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \otimes \boldsymbol{q}_{\mathrm{B/A}} - \frac{1}{2}\boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{q}^{*}_{\mathrm{B/A}} \otimes (\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} - \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}) \otimes \boldsymbol{q}_{\mathrm{B/A}} \tag{4.73}$$

Therefore, the relative attitude kinematics equation becomes,

$$\dot{\boldsymbol{q}}_{\mathrm{B/A}} = \frac{1}{2}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \otimes \boldsymbol{q}_{\mathrm{B/A}} \tag{4.74}$$

For relative attitude dynamics we need to differentiate the Eq. (4.68),

$$\dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/A}} = \dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}} - \boldsymbol{C}_{\mathrm{B/A}}\dot{\boldsymbol{\omega}}^{\mathrm{A}}_{\mathrm{A/I}} + \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \times \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \tag{4.75}$$

Substituting for $\dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}}$ from the attitude dynamics given in Eq. (4.58),

$$\begin{aligned}
\dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/A}} &= \boldsymbol{J}^{-1}(\boldsymbol{T}^{\mathrm{B}} - \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}) - \boldsymbol{C}_{\mathrm{B/A}}\dot{\boldsymbol{\omega}}^{\mathrm{A}}_{\mathrm{A/I}} + \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \times \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \\
&= \boldsymbol{J}^{-1}(\boldsymbol{T}^{\mathrm{B}} - (\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} + \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}}) \times \boldsymbol{J}(\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} + \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}})) - \boldsymbol{C}_{\mathrm{B/A}}\dot{\boldsymbol{\omega}}^{\mathrm{A}}_{\mathrm{A/I}} + \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \times \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \\
&= \boldsymbol{J}^{-1}(\boldsymbol{T}^{\mathrm{B}} - \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} - \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}} \times \boldsymbol{J}\boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}}) + \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \times \boldsymbol{C}_{\mathrm{B/A}}\boldsymbol{\omega}^{\mathrm{A}}_{\mathrm{A/I}}
\end{aligned} \tag{4.76}$$

The relative attitude dynamic equation is therefore, given as below.

$$\dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/A}} = \boldsymbol{J}^{-1}(\boldsymbol{T}^{\mathrm{B}} - \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} - \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{A/I}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{A/I}}) - \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{A/I}} \times \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \tag{4.77}$$

4.9. Dual Quaternion Dynamics and Kinematics

To implement the inertial and relative kinematics and dynamics in the guidance algorithms, we need to convert the EOMs to the form of dual quaternions. In this section, we derive the complete inertial and relative EOMS using DQs.

4.9.1. DQ Inertial EOMS

The derivations of the kinematic and dynamic equations of the SC in dual quaternion form can be found in Lee and Mesbahi (2015) and Filipe and Tsiotras (2013). They kinematic equation can be given as,

$$\dot{\check{\boldsymbol{q}}}_{\mathrm{B/I}} = \frac{1}{2}\check{\boldsymbol{q}}_{\mathrm{B/I}} \check{\otimes} \check{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}} \tag{4.78}$$

where, the dual quaternion and dual velocity are as below,

$$\check{\boldsymbol{q}}_{\mathrm{B/I}} = \boldsymbol{q}_{\mathrm{B/I}} + \frac{1}{2}\boldsymbol{r}^{\mathrm{I}} \odot \boldsymbol{q}_{\mathrm{B/I}}\epsilon, \quad \check{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}} = \begin{pmatrix} \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \\ \boldsymbol{v}^{\mathrm{B}}_{\mathrm{B/I}} \end{pmatrix}$$

The dynamic equation is, as follows,

$$\check{\boldsymbol{J}}\dot{\check{\boldsymbol{\omega}}}^{\mathrm{B}}_{\mathrm{B/I}} + \check{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}} \check{\times} \check{\boldsymbol{J}}\check{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}} = \check{\boldsymbol{F}}^{\mathrm{B}} \tag{4.79}$$

where, the matrix for inertia, $\boldsymbol{J}$ and mass, m combined and that for force, $\boldsymbol{F}$ and torque, $\boldsymbol{T}$ combined are,

$$\check{\boldsymbol{J}} = \left[\begin{array}{cc|cc} \boldsymbol{0}_{3\times3} & 0 & m\boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & 1 \\ \hline \boldsymbol{J} & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 1 & 0 & 0 \end{array}\right], \quad \check{\boldsymbol{F}}^{\mathrm{B}} = \begin{pmatrix} \boldsymbol{F}^{\mathrm{B}} \\ 0 \\ \boldsymbol{T}^{\mathrm{B}} \\ 0 \end{pmatrix}$$

with the translational and rotational dynamic EOMs, as follows,

$$\begin{aligned} \boldsymbol{F}^{\mathrm{B}} &= m\dot{\boldsymbol{v}}^{\mathrm{B}}_{\mathrm{B/I}} + \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \times m\boldsymbol{v}^{\mathrm{B}}_{\mathrm{B/I}} \\ \boldsymbol{T}^{\mathrm{B}} &= \boldsymbol{J}\dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}} + \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \end{aligned}$$

These equations will be used to represent the SC orbital and attitude dynamics in the inertial frame fixed to the asteroid centre while developing the G&C algorithms using dual quaternions. These equations can be converted to obtain the relative state of the SC with respect to the rotating centre fixed frame at the target body.

4.9.2. DQ Relative Kinematics

The DQ representation of the body frame, B with respect to the asteroid rotating frame, A is $\check{\boldsymbol{q}}_{\mathrm{B/A}} = \boldsymbol{q}_{\mathrm{B/A}} + \frac{\epsilon}{2}\boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{r}^{\mathrm{A}}$ with reference to discussions in Chapter 3. Differentiating this with respect of time we get,

$$\begin{aligned} \dot{\check{\boldsymbol{q}}}_{\mathrm{B/A}} &= \dot{\boldsymbol{q}}_{\mathrm{B/A}} + \frac{\epsilon}{2}\left(\dot{\boldsymbol{q}}_{\mathrm{B/A}} \otimes \boldsymbol{r}^{\mathrm{A}} + \boldsymbol{q}_{\mathrm{B/A}} \otimes \dot{\boldsymbol{r}}^{\mathrm{A}}_{\mathrm{B/A}}\right) \\ &= \frac{1}{2}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \otimes \boldsymbol{q}_{\mathrm{B/A}} + \frac{\epsilon}{2}\left(\frac{1}{2}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} \otimes \boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{r}^{\mathrm{A}} + \boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}}\right) \end{aligned} \tag{4.80}$$

Rearranging Eq. (4.80) in the matrix form we get,

$$\dot{\check{\boldsymbol{q}}}_{\mathrm{B/A}} = \frac{1}{2}\begin{bmatrix} [\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] & 0_{4\times4} \\ [\boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}}\odot] & [\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] \end{bmatrix}\begin{pmatrix} \boldsymbol{q}_{\mathrm{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{r}^{\mathrm{A}} \end{pmatrix} \tag{4.81}$$

For convenience, representing the velocity in the B frame we get

$$\boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}} = \boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{v}^{\mathrm{A}}_{\mathrm{B/A}} \otimes \boldsymbol{q}^{*}_{\mathrm{B/A}} \otimes \boldsymbol{q}_{\mathrm{B/A}} = \boldsymbol{v}^{\mathrm{B}}_{\mathrm{B/A}} \otimes \boldsymbol{q}_{\mathrm{B/A}} \tag{4.82}$$

Therefore, Eq. (4.81) can be rewritten as,

$$\dot{\check{\boldsymbol{q}}}_{\mathrm{B/A}} = \frac{1}{2}\begin{bmatrix} [\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] & 0_{4\times4} \\ [\boldsymbol{v}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] & [\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] \end{bmatrix}\begin{pmatrix} \boldsymbol{q}_{\mathrm{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathrm{B/A}} \otimes \boldsymbol{r}^{\mathrm{A}} \end{pmatrix} \tag{4.83}$$

Introducing a new variable as dual velocity,

$$\check{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/A}} = \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}} + \epsilon\boldsymbol{v}^{\mathrm{B}}_{\mathrm{B/A}} \tag{4.84}$$

Therefore, the combined translational and rotational kinematics can be represented in the DQ form, as follows,

$$\dot{\check{\boldsymbol{q}}}_{\mathrm{B/A}} = \frac{1}{2}\check{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/A}} \check{\otimes} \check{\boldsymbol{q}}_{\mathrm{B/A}} \tag{4.85}$$

where, $\check{\otimes}$ represents the DQ product and can be written in the form,

$$[\check{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/A}}\check{\otimes}] = \begin{bmatrix} [\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] & 0_{4\times4} \\ [\boldsymbol{v}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] & [\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/A}}\otimes] \end{bmatrix} \tag{4.86}$$

4.9.3. DQ Relative Dynamics

Eqs. (4.66) and (4.77), give the formulation for relative translational and rotational dynamics. Reiterating them here,

$$\begin{aligned}\boldsymbol{a}^{\mathtt{A}}_{\mathtt{B/A}} &= \boldsymbol{C}_{\mathtt{A/I}}\ddot{\boldsymbol{r}}^{\mathtt{I}} - 2\boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}} - \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{r}^{\mathtt{A}} \\ \dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} &= \boldsymbol{J}^{-1}(\boldsymbol{T}^{\mathtt{B}} - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}) + \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \times \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}\end{aligned} \tag{4.87}$$

where, the translational dynamics can be rewritten as,

$$\boldsymbol{a}^{\mathtt{A}}_{\mathtt{B/A}} = \left(\frac{\boldsymbol{F}^{\mathtt{A}}}{m} + \boldsymbol{g}^{\mathtt{A}}\right) - 2\boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}} - \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{r}^{\mathtt{A}} \tag{4.88}$$

Now, since the dual velocity is in the B frame, we need to convert the translational dynamics to B frame

$$\boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} = \boldsymbol{C}_{\mathtt{B/A}}\boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}} \tag{4.89}$$

Differentiating Eq. (4.89), the translational dynamics in the B frame can therefore, be given as,

$$\begin{aligned}\boldsymbol{a}^{\mathtt{B}}_{\mathtt{B/A}} &= \boldsymbol{C}_{\mathtt{B/A}}\dot{\boldsymbol{v}}^{\mathtt{A}}_{\mathtt{B/A}} - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \times \boldsymbol{C}_{\mathtt{B/A}}\boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}} \\ \boldsymbol{a}^{\mathtt{B}}_{\mathtt{B/A}} &= \boldsymbol{C}_{\mathtt{B/A}}\left(\frac{\boldsymbol{F}^{\mathtt{A}}}{m} + \boldsymbol{g}^{\mathtt{A}} + \boldsymbol{a}^{\mathtt{A}}_{D} - 2\boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}} - \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} \times \boldsymbol{r}^{\mathtt{A}}\right) - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \times \boldsymbol{C}_{\mathtt{B/A}}\boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}}\end{aligned} \tag{4.90}$$

Rearranging Eqs. (4.90) and (4.77), respectively, we get,

$$\boldsymbol{F}^{\mathtt{B}} + m\boldsymbol{g}^{\mathtt{B}} + m\boldsymbol{a}^{\mathtt{B}}_{D} = m\dot{\boldsymbol{v}}^{\mathtt{B}}_{\mathtt{B/A}} + \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \times m\boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} + \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \times \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \times m\boldsymbol{r}^{\mathtt{B}} + 2m\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \times \boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} \tag{4.91}$$

$$\boldsymbol{r}_{ex} \times \boldsymbol{F}^{\mathtt{B}} + \boldsymbol{T}^{\mathtt{B}}_{D} = \boldsymbol{J}\dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + (\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} + \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}) \times \boldsymbol{J}(\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} + \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}) + \boldsymbol{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \times \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \tag{4.92}$$

For DQ dynamics, we need to combine Eqs. (4.91) and (4.92), such that it incorporates the dual velocity defined in the earlier subsection of DQ relative kinematics. using the following matrix representations.

$$\check{\boldsymbol{F}}^{\mathtt{B}} = \begin{pmatrix} \boldsymbol{F}^{\mathtt{B}} \\ 0 \\ \boldsymbol{r}_{ex} \times \boldsymbol{F}^{\mathtt{B}} \\ 0 \end{pmatrix}, \quad \check{\boldsymbol{F}}^{\mathtt{B}}_{D} = \begin{pmatrix} m\boldsymbol{g}^{\mathtt{B}} + m\boldsymbol{a}^{\mathtt{B}}_{D} \\ 0 \\ \boldsymbol{T}^{\mathtt{B}}_{D} \\ 0 \end{pmatrix}, \quad \check{\boldsymbol{J}} = \left[\begin{array}{cc|cc} \boldsymbol{0}_{3\times3} & 0 & m\boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & 1 \\ \hline \boldsymbol{J} & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \tag{4.93}$$

After some rearranging, we can have the dynamics in the form,

$$\check{\boldsymbol{F}}^{\mathtt{B}} + \check{\boldsymbol{F}}^{\mathtt{B}}_{D} = \check{\boldsymbol{J}}\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}} + (\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}})\check{\times}\check{\boldsymbol{J}}(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}) + \check{\boldsymbol{J}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{J}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\boldsymbol{R} \tag{4.94}$$

where,

$$\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}} = \begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \\ \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}} \end{pmatrix}, \quad [\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}] = \begin{bmatrix} [\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}\times] & 0_{4\times4} \\ \\ [\boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}}\times] & [\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}\times] \end{bmatrix} \tag{4.95}$$

$\boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}}$ is zero, since the origins of the frames A and I are the centre of the asteroid. The matrix $\boldsymbol{R}$ is given by the quaternion product from DQ properties,

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{0}_{4\times1} \\ 2\boldsymbol{q}_d \otimes \boldsymbol{q}^{*}_{\mathtt{B/A}} \end{pmatrix} \tag{4.96}$$

4.10. Internal Disturbances

Internal disturbing forces acting on the SC, include the accelerations due to unwanted oscillations of flexible appendages and sloshing of propellant. Flexible appendages include the solar arrays and antennas onboard the SC. The increasing demand of power for sophisticated onboard instruments, to carry out scientific experiments and communications, has led to an increase in the size of the solar arrays (SAs) onboard the SC. Also, at a distance of 1.2 to 3 au, the solar arrays and antennas have to be massive, to accommodate for the lowered solar irradiance and distance from Earth, respectively. The need for appendages to be larger in size and more in number, leads to the requirement of using lightweight material, which in turn increases their flexibility. They are, therefore, subject to unwanted oscillations when intended or unintended attitude changes occur.

Grandsen and Mooij (2018) have proved in their research, that a simple adaptive controller can be used to stabilise an SC similar to Rosetta, under the impact of debris on the solar panel, causing unmodelled disturbances. They also suggest means, by which robust performance can be increased for possible scenarios other than debris impact. Flexible dynamics of the SC, is therefore, not considered for modelling while developing the guidance algorithms for this thesis.

Approximately 50% mass of the SC is contributed by the propellant and this causes non-negligible sloshing effects of the SC dynamics. But, modelling slosh dynamics requires structural details of the propellant tank and also exact volumes of propellant remaining in the tank before initialisation of the mission phases. Usage of a bi-propellant tank further increases the complexity of modelling.

Since, the modelling of internal perturbations is additional to the effort required to develop the guidance algorithms using dual quaternions, it does not fit the schedule. The main objective of this thesis is to develop a guidance algorithm, which can be satisfied without the inclusion of these models and hence will be left for future implementation.

4.11. External Disturbances

External disturbances on the SC for an asteroid mission include thermal recoil forces, solar radiation pressure, third body perturbations, gravity gradient torque and mass expulsion forces. Mass expulsion forces will be excluded from the analysis of external disturbances, since their sources can be identified by experimentation during the verification and validation testing of SC subsystems before the mission. Modelling of the thermal recoil forces, requires knowledge of specific SC structural and material design. Yoshihide et al. (2010) have modelled the thermal recoil forces for Rosetta and found it to be ≈5 - 8% of SRP disturbances, which proves that their magnitude is very less as compared to the other perturbances. This leaves us with the three major contributors to the accelerations perturbing the SC motion.

Figure 4.5, plots these accelerations for Rosetta while performing close proximity operations about Comet 67P. The mission phases during these periods is given in Table 4.1. Coma drag can be ignored, since we are developing the algorithm for an asteroid mission. As can be seen from the figure the accelerations from the central gravity field is the highest, its maximum being $\approx 10^{-8}$m/s^2. But in case of lower gravitational acceleration, it can be seen that the accelerations due to SRP, contribute to just a factor of 10 lesser. This makes it a requirement to model the SRP along with the gravitational field of the target asteroid. Third body perturbations due to the Sun, however, can be seen to have the lowest magnitude, $\approx 10^3$ to 10^4 times lower than central gravity and SRP.

Table 4.1: Rosetta mission phases from August 2014 to November 2014 (Godard et al. (2015)).

Phase Name	Start Date
Close Approach Trajectory	2014/08/01
Transition to Global Mapping Phase	2014/09/02
Global Mapping Phase	2014/09/09
Close Observation Phase	2014/10/06
Science surface package Delivery Phase	2014/10/24

During the global mapping, close observation and surface package delivery phase, it can be seen, that the second degree gravitational acceleration of the comet peaks and is higher than SRP. This proves to be a baseline for the asteroid mapping and descent scenario as well, since the mission profiles are similar. The irregular shape, non-spherical mass distribution and rotational state of a small Solar System bodies accounts for the variability in the gravitational acceleration, as can be seen in the plots. This calls for a gravity model that takes into account the shape of the asteroid rather than a central gravity field. Using a central gravity field or a flat asteroid model will induce errors in the trajectories to be found by the optimiser, especially for the mapping phase. This also concerns building an algorithm for safe guidance of the SC, such that it can handle variational gravity field and perturbances in real applications.

The SC chosen for this thesis is Rosetta for mapping and the SC solar panels reduced in size for the descent phase. For both phases, the SC is still symmetric about each axis, geometrically and with surface properties. Hence, the torque due to SRP can be ignored. Due to the non-uniform gravity field as well as the orientation of the SC with respect to the small body, different parts of the SC structure will be subjected to different gravitational attractive forces. The SC is, therefore, subjected to gravity gradient torque, which could destabilise

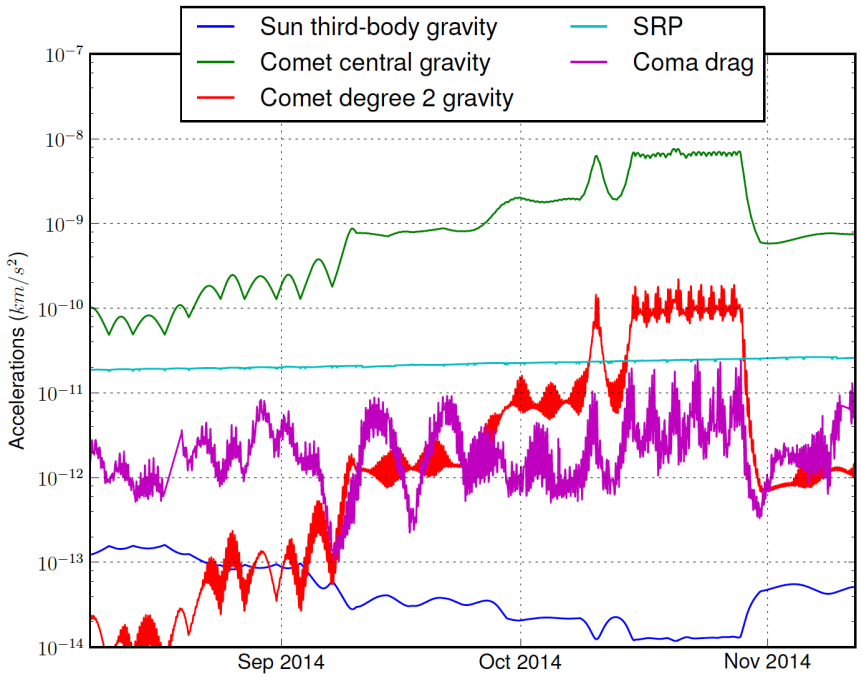


Figure 4.5: Gravitational and perturbing accelerations on the SC Rosetta, while performing close proximity operations (Godard et al. (2015)).

it, if unaccounted for.

In the next sections, the mathematical models of variational gravity field, gravity gradient torque, SRP and 3rd body perturbation have been provided. The 3rd body perturbation has been implemented, just to verify that it's magnitude is lower than SRP for the asteroid Kleopatra, which is at 2.79 au. Figure 7.19 in Chapter 7 presents the results for a specified orbit, that proves it is lower than both SRP and gravitational accelerations and hence will not be included in the environmental models for the algorithm.

4.11.1. Small Body Gravity Field

Modelling the gravitational field of an irregular shaped small body is a challenging astrodynamics problem. There are three fundamentally different ways of approaching this problem. The first approach involves using a series expansion to approximate the gravitational potential for example using spherical or ellipsoidal harmonics (Takahashi and Scheeres (2014)). In this approach, series coefficients are evaluated as integrals over the volume of the body and in case of a constant density body, it is reduced to integrals over just the surface of the body. This approach has more disadvantages due to higher truncation error when the gravity field is evaluated close to the model's radius of convergence, inside the Brillouin sphere[1] for highly irregular shaped asteroids. Also this approach cannot say whether the field point is inside or outside the body and a separate algorithm is required to detect this crucial geometric information.

The second approach is mass concentration where, the target body is filled with point masses on an evenly spaced grid till the total mass of the body is realised (Geissler et al. (1996)). This method proves to be less accurate than the harmonics approach, even though it does not diverge and will converge to the true gravity field for infinite number of mass points. It is, therefore, limited by the number of point masses, since it is computationally difficult and still has significant errors with large numbers of mascons. Also, it does not provide information, whether, the field point in inside or outside the body, same as the harmonics approach.

The third approach is the constant density polyhedron, where, direct computation of the gravitational potential is done using a finite number of polyhedral definitions (Werner and Scheeres (1997)). For the given shape and density this gravity field is exact and the errors can be reduced to the shape determination and level of chosen discretisation. This model is valid up to the surface of the body and has no region of divergence. It, therefore, helps in studying the dynamics of particles launched from the surface. A polyhedron algorithm

[1] A Brillouin sphere is a minimum sphere, that circumscribes the body

can detect whether the field point is inside or outside the polyhedron by evaluating the Laplacian of the gravitational potential. Since, we deal with close proximity operations, the polyhedron approach suits the mission best and is used for this thesis. The gravitational potential of a constant density polyhedron at a field point $\boldsymbol{r}$ is given as follows,

$$U_p(\mathbf{r}) = \frac{1}{2} G\varrho \sum_{e \in edges} \boldsymbol{r}_e^T \boldsymbol{E}_e \boldsymbol{r}_e \cdot L_e - \frac{1}{2} G\varrho \sum_{f \in faces} \boldsymbol{r}_f^T \boldsymbol{F}_f \boldsymbol{r}_f \cdot \omega_f \tag{4.97}$$

where, G is the gravitational constant, ϱ is the mean asteroid density (assumed to be constant), r_e and r_f are distances from a field point to an edge and a face, respectively, E_e is a dyadic matrix of an edge and F_f is an outer product of a face normal. They can be calculated as follows,

$$\boldsymbol{E}_e = \boldsymbol{n}_A (\boldsymbol{n}_{12}^A)^T + \boldsymbol{n}_B (\boldsymbol{n}_{21}^B)^T \tag{4.98}$$

$$\boldsymbol{F}_f = \boldsymbol{n}_f (\boldsymbol{n}_f)^T \tag{4.99}$$

where, $\boldsymbol{n}_A$, $\boldsymbol{n}_B$ and $\boldsymbol{n}_f$ are unit vectors normal to the faces A, B and a general face, respectively. The unit vectors $\boldsymbol{n}_{12}^A$ and $\boldsymbol{n}_{21}^B$ are normal to the edge. They are associated with the face normal. These vectors are represented in the Figure 4.6a. Using the position vectors from the asteroid frame to the face's vertices as shown in Figure 4.6b, these normals can be calculated as follows,

$$\boldsymbol{n}_f = \frac{(\boldsymbol{r}_2 - \boldsymbol{r}_1) \times (\boldsymbol{r}_3 - \boldsymbol{r}_2)}{\|(\boldsymbol{r}_2 - \boldsymbol{r}_1) \times (\boldsymbol{r}_3 - \boldsymbol{r}_2)\|} \tag{4.100}$$

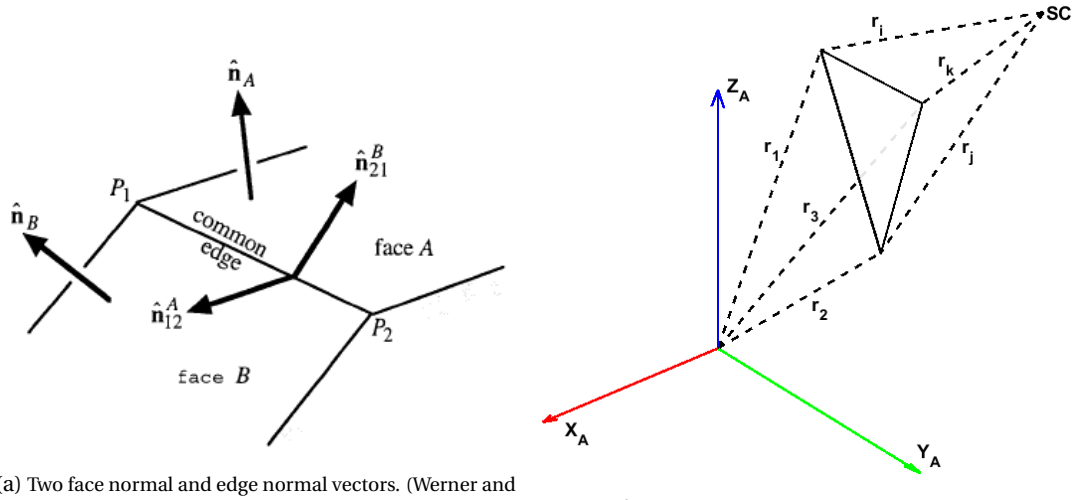
$$\begin{aligned} \boldsymbol{n}_{12}^A &= \frac{(\boldsymbol{r}_1 - \boldsymbol{r}_2) \times \boldsymbol{n}_A}{\|(\boldsymbol{r}_1 - \boldsymbol{r}_2) \times \boldsymbol{n}_A\|} \\ \boldsymbol{n}_{21}^B &= \frac{(\boldsymbol{r}_2 - \boldsymbol{r}_1) \times \boldsymbol{n}_B}{\|(\boldsymbol{r}_2 - \boldsymbol{r}_1) \times \boldsymbol{n}_B\|} \end{aligned} \tag{4.101}$$

The vectors from the SC/field point to the vertices of the face ($\boldsymbol{r}_i$, $\boldsymbol{r}_j$, $\boldsymbol{r}_k$) can be defined as follows,

$$\begin{aligned} \boldsymbol{r}_i &= \boldsymbol{r}_1 - \boldsymbol{r}_A \\ \boldsymbol{r}_j &= \boldsymbol{r}_2 - \boldsymbol{r}_A \\ \boldsymbol{r}_k &= \boldsymbol{r}_3 - \boldsymbol{r}_A \end{aligned} \tag{4.102}$$

where, $\boldsymbol{r}_A$ is the position vector of the SC/field point in the asteroid frame. The vectors have magnitudes r_i, r_j, r_k, respectively. Using these definitions the dimensionless factors L_e and ω_f, which are the potential of a wire and the pre-face factor, respectively are given as below,

$$L_e = ln \frac{r_i + r_j + e_{ij}}{r_i + r_j - e_{ij}} \tag{4.103}$$



(a) Two face normal and edge normal vectors. (Werner and Scheeres (1997))

(b) Position vectors to the vertices of a face.

Figure 4.6: An edge at two faces.

$$\omega_f = 2\arctan\frac{\boldsymbol{r}_i \cdot \boldsymbol{r}_j \times \boldsymbol{r}_k}{r_i r_j r_k + r_i(\boldsymbol{r}_j \cdot \boldsymbol{r}_k) + r_j(\boldsymbol{r}_k \cdot \boldsymbol{r}_i) + r_k(\boldsymbol{r}_i \cdot \boldsymbol{r}_j)} \tag{4.104}$$

The vectors $\boldsymbol{r}_e$ and $\boldsymbol{r}_f$ from Eq. (4.97) are arbitrary vectors from the SC to an edge and a face, respectively. They are defined as follows,

$$\boldsymbol{r}_e = \frac{\boldsymbol{r}_i + \boldsymbol{r}_j}{2} \tag{4.105}$$

$$\boldsymbol{r}_f = \frac{\boldsymbol{r}_i + \boldsymbol{r}_j + \boldsymbol{r}_k}{3} \tag{4.106}$$

such that they lie in the middle of an edge and a triangular face. The potential gradient or the acceleration due to gravity (g^{A}) and the Laplacian of the potential, which can be used in the know whether the asteroid would crash into the asteroid are then given as follows,

$$\boldsymbol{g}^{\mathsf{A}} = \nabla U = -G\,\varrho \sum_{e \in edges} \boldsymbol{E}_e\,\boldsymbol{r}_e\,L_e + G\,\varrho \sum_{f \in faces} \boldsymbol{F}_f\,\boldsymbol{r}_f\,\omega_f \tag{4.107}$$

$$\nabla^2 U = -G\,\varrho \sum_{f \in faces} \omega_f \tag{4.108}$$

Computing the Laplacian provides information on whether the field point is inside, outside, on a face, on a vertex or on an edge (Park et al. (2010)).

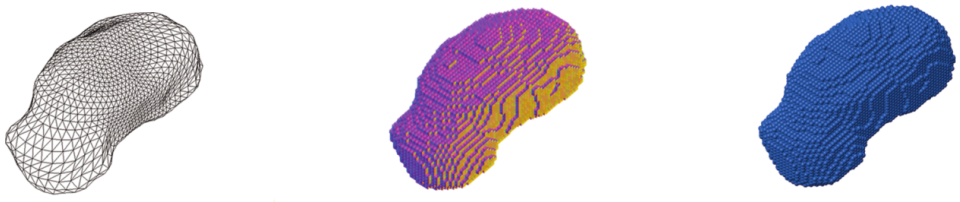
$$-\frac{\nabla^2 U(\boldsymbol{r}_i)}{G\,\varrho} = \begin{cases} 4\pi, & \text{if inside} \\ 0, & \text{if outside} \\ 2\pi, & \text{if on a face} \\ \text{solid angle}, & \text{if on a vertex or on an edge} \end{cases} \tag{4.109}$$

The disadvantage of this method is, that it assumes a constant density of the asteroid for the computation and therefore, for an object with significant density variation this is not a good model. Since, even with constant density, polyhedral gravity computations are expensive, it is not enhanced to accommodate varying density models. Hence, just the surface finite element model, as shown in Figure 4.7a is used for this thesis.

For future implementation, a finite element approach has been developed by Park et al. (2010), to overcome this disadvantage. Optical survey of the asteroid provides a high precision polyhedral shape, which is then filled with finite elements like cubes or spheres with distinct density values. Figs. 4.7b and 4.7c provide the predicted shape model of Itokawa from radar survey and constructed shape models filled with finite elements like cubes and spheres, respectively. Once the constructed models are available, the attraction saying whether the SC is inside the Brillouin sphere or the body has a significant density variation can be computed. The true attraction from the polyhedron model is then compared with that based on the finite element model and it is found, that the later provides a very good approximation of the polyhedral shape model. But these models increase the computations costs tremendously and can become an ill-conditioned problem based on the finite element size.

Gravity Gradient Torque

The gravity gradient torque can be found by summing the contributions of the gravitational field on various point masses of the SC. For a central gravitational field, that is spherically symmetric and the SC as a rigid



(a) Polyhedron model (3688 faces)

(b) Finite-cube shape model (26311 elements)

(c) Finite-sphere shape model (26311 elements)

Figure 4.7: Different types of shape models for Itokawa (Park et al. (2010))

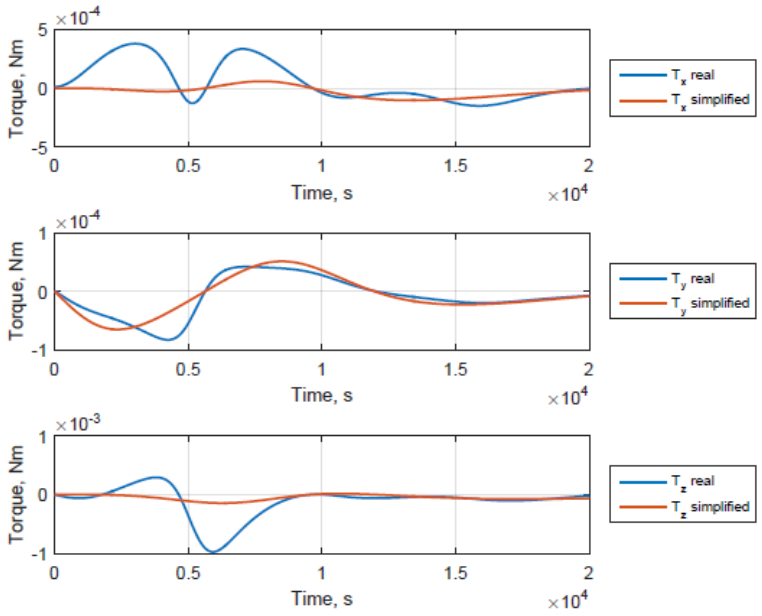


Figure 4.8: Gravity gradient torques from central gravity and polyhedron models compared in Kleopatra's gravity field for Rosetta. (Razgus (2017))

body, the force on any i^{th} point mass of the SC having n number of point masses, can be given by (Wie (2001)),

$$\boldsymbol{F}_{GG_i} = m_i \boldsymbol{g}_{C_i} = -\frac{\mu \boldsymbol{r}_i \, m_i}{|\boldsymbol{r}_i|^3} \tag{4.110}$$

where, $\boldsymbol{g}_{C_i}$ is the gravitational acceleration at i^{th} point due to the central gravity field, $\boldsymbol{r}_i$ is the position vector of the i^{th} point mass with respect to the asteroid reference frame and m_i is the mass of the i^{th} point mass. The gravity gradient torque can therefore, be given as,

$$\boldsymbol{T}_{GG} = \sum_{i=1}^{n} \boldsymbol{\rho}_i \times \boldsymbol{F}_{GG_i} = -\mu \sum_{i=1}^{n} \frac{\boldsymbol{\rho}_i \times (\boldsymbol{r}_c + \boldsymbol{\rho}_i)}{|\boldsymbol{r}_c + \boldsymbol{\rho}_i|^3} m_i = -\mu \sum_{i=1}^{n} \frac{\boldsymbol{\rho}_i \times \boldsymbol{r}_c}{|\boldsymbol{r}_c + \boldsymbol{\rho}_i|^3} m_i \tag{4.111}$$

where, $\boldsymbol{\rho}_i$ is the position vector of the i^{th} mass point with respect to the SC's COM and $\boldsymbol{r}_c$ is the vector between the COM of the asteroid and the SC. Using the expansion of $|(\boldsymbol{r}_c + \boldsymbol{\rho}_i)|^{-3}$ series and approximating it to the first and second order terms we get,

$$\begin{aligned} |\boldsymbol{r}_c + \boldsymbol{\rho}_i|^{-3} &= r_c^{-3} \left\{ 1 + \frac{2(\boldsymbol{r}_c \cdot \boldsymbol{\rho}_i)}{r_c^2} + \frac{\boldsymbol{\rho}_i^2}{r_c^2} \right\}^{-\frac{3}{2}} \\ &= r_c^{-3} \left\{ 1 - \frac{3(\boldsymbol{r}_c \cdot \boldsymbol{\rho}_i)}{r_c^2} \right\} \end{aligned} \tag{4.112}$$

Resubstituting in Eq. (4.111), we get,

$$\boldsymbol{T}_{GG} = -\frac{3\mu}{r_c^5} \sum_{i=1}^{n} (\boldsymbol{r}_c \cdot \boldsymbol{\rho}_i)(\boldsymbol{\rho}_i \times \boldsymbol{r}_c) m_i \tag{4.113}$$

Further manipulation and simplification of the equation gets us to the result,

$$\boldsymbol{T}_{GG} = -\frac{3\mu}{r_c^5} \boldsymbol{r_c} \times \boldsymbol{J}.\boldsymbol{r}_c \tag{4.114}$$

where, $\boldsymbol{J}$ is the inertia moment tensor of the SC. This equation has some simplifications and assumptions, that make it unsuitable to be used for asteroids. It assumes a central gravity field with spherical symmetry and also neglects the higher order terms of the gravity field. The polyhedron model presented in the previous

section provides a better model for the gravity gradient torque for asteroids. In the report by Razgus (2017), the gravity gradient torque achieved from the central gravity field and polyhedron field have been compared and plotted for Kleopatra as can be seen in Figure 4.8. It can be seen from the figure, that the central gravity field fails to incorporate higher order couplings and/or non central gravity field terms. Even though a similar trend is observed the magnitudes are way off.

Using the result from the polyhedron model for the acceleration due to gravity in Eq. (4.107) and transforming it to the body frame, B, we can rewrite the torque Eq. (4.111) as,

$$\boldsymbol{T}^{\mathrm{B}}_{GG} = \sum_{i=1}^{n} \boldsymbol{\rho}_i \times \boldsymbol{F}^i_{GG} = \sum_{i=1}^{n} \boldsymbol{\rho}_i \times m_i \, \boldsymbol{g}^{\mathrm{B}}_i \tag{4.115}$$

where, $\boldsymbol{g}^{\mathrm{B}}_i$ represents the gravitational acceleration at the i^{th} point mass from the polyhedron model. Razgus (2017) has also presented the comparison between GG torques using the polyhedron acceleration due to gravity for an increasing number of point masses. Relative errors between the models are compared using the formula,

$$T = \frac{T_{real} - T_N}{T_{real}} \times 100\% \tag{4.116}$$

where, T_{real} represents the most accurate representation of the GG torques and T_N represents the GG torques from the lower value of point masses model of the SC. Another observation made from these empirical calculations is, that the error between the real torque model and another lower-numbered point mass model remains constant throughout. Although it seems intuitive it lacks background theory as to why it should remain constant. But this proves useful, since once the comparison has been made, computations can be reduced to the lower point masses model and it can be simply corrected for the constant error.

4.11.2. Solar Radiation Pressure

SRP is another contributor to the disturbance torque on the SC. This force is, of course, zero when the SC is in the shadow of the target body. For most applications, onboard movable surfaces are commanded from the ground station rather than autonomous real-time control by the onboard computer. For this, the SC's surface is modelled to a collection of N plates of a particular surface area A, having specular reflection coefficient as α_r, diffuse reflection coefficient as α_d and absorption coefficient α_a. Let $e_\odot$ be the vector from the plate to the Sun, n be the normal vector from the plate and r be the position vector from the com of the SC to the point of pressure on the plate, all in the B frame. The angle between the normal vector from the plane and the vector from the plane to the Sun is given by,

$$\cos\,\theta = \boldsymbol{n} \cdot \boldsymbol{e}_\odot \tag{4.117}$$

With this the force due to SRP acting on individual illuminated plates on the SC (Markley and Crassidis (2013)) is as follows,

$$\boldsymbol{F}^i_{SRP} = -P_\odot A_i \cos\,\theta_i \left(2\left(\frac{\alpha^i_d}{3} + \alpha^i_r \cos\,\theta_i\right)\boldsymbol{n}_i + (1-\alpha^i_r)\boldsymbol{e}_{\odot,i}\right) \tag{4.118}$$

where, $P_\odot$ is the solar radiation pressure. The acceleration and total torque acting on the SC due to this force can be further given by,

$$F^{\mathrm{B}}_{SRP} = \sum_{i=1}^{N} \boldsymbol{F}^i_{SRP} \tag{4.119}$$

$$\boldsymbol{T}^{\mathrm{B}}_{SRP} = \sum_{i=1}^{N} \boldsymbol{r}_i \times \boldsymbol{F}^i_{SRP} \tag{4.120}$$

4.11.3. Third Body Gravitational Perturbation

A celestial body or object is under the influence of gravitational attraction of other celestial bodies. The gravitational forces of bodies besides the central body leads to perturbation in the motion of the body under their influence. In case of an SC revolving about an asteroid, this could cause a significant disturbance due to the SC's low mass as well as the weak and irregular gravity field of the asteroid. The magnitude of this force varies depending on the distance of the perturbing celestial body as well as its mass, as is known from the Newtonian gravitational attraction. In the Solar System, the Sun proves as the biggest source of perturbing accelerations. Since, we are considering NEAs and asteroids in the MAB, the gravitational forces of Earth,

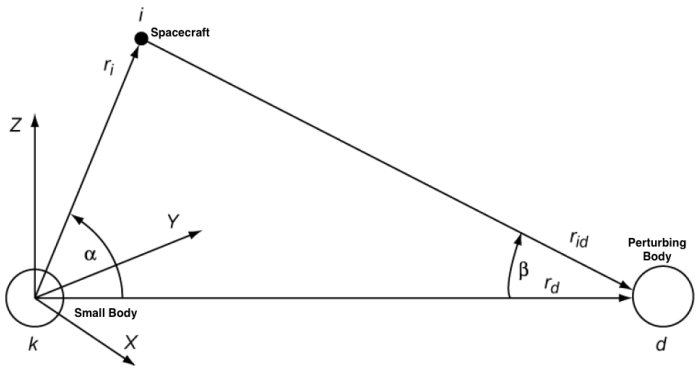


Figure 4.9: Relative positions of the SC, small body and the perturbing body. (Wakker (2015))

Mars or Jupiter could also prove to be significant perturbing forces. In Figure 4.9 the relative positions of the asteroid as the central body, the SC and the perturbing body with the respect to the asteroid reference frame have been represented.

For a single perturbing body, we can treat the system as a three-body problem to find the relative acceleration of the SC with respect to the asteroid due to the perturbing force. The total acceleration of the SC can be given as (Wakker (2015)),

$$\boldsymbol{a}_T = \frac{\mathrm{d}^2 \boldsymbol{r}_i}{\mathrm{dt}^2} = -G\frac{m_i + m_k}{r_i^3}\boldsymbol{r}_i + G\sum_{d\neq i,k} m_d\left(\frac{\boldsymbol{r}_d - \boldsymbol{r}_i}{r_{id}^3} - \frac{\boldsymbol{r}_d}{r_d^3}\right) \tag{4.121}$$

where, m_i, m_d, m_k are the masses of the SC, perturbing body and asteroid, respectively and r_i, r_d are the position vectors of the SC and the perturbing body with respect to the asteroid reference frame, respectively. The first term on the right hand side of the Eq. (4.121) represents the acceleration due to a central gravity field of the asteroid and the second term represents the acceleration due to the perturbing body. The perturbing acceleration due to a single body is therefore,

$$\begin{aligned}\boldsymbol{a}_P &= -G\, m_d\left(\frac{\boldsymbol{r}_d - \boldsymbol{r}_i}{r_{id}^3} - \frac{\boldsymbol{r}_d}{r_d^3}\right)\\ &= -G\, m_d\sqrt{\left(\frac{\boldsymbol{r}_{id}}{r_{id}^3} - \frac{\boldsymbol{r}_d}{r_d^3}\right)\cdot\left(\frac{\boldsymbol{r}_{id}}{r_{id}^3} - \frac{\boldsymbol{r}_d}{r_d^3}\right)}\\ &= -G\, m_d\sqrt{\frac{1}{r_{id}^4} + \frac{1}{r_d^4} + \frac{2\cos\beta}{r_{id}^2 r_d^2}}\end{aligned} \tag{4.122}$$

From the Figure 4.9 we get,

$$\cos\beta = \frac{r_d - r_i\cos\alpha}{r_{id}};\quad r_{id}^2 = r_i^2 + r_d^2 - 2r_i r_d\cos\alpha \tag{4.123}$$

Using Eq. (4.123) and substituting in Eq. (4.122),

$$\boldsymbol{a}_P = -G\frac{m_d}{r_d^2}\sqrt{1 + \frac{1}{(1-2\gamma\cos\alpha+\gamma^2)^2} - \frac{2(1-\gamma\cos\alpha)}{(1-2\gamma\cos\alpha+\gamma^2)^{3/2}}} \tag{4.124}$$

where, $\gamma = r_i/r_d$. Now for the case of an SC orbiting a celestial body, $r_d > r_i$ and therefore, γ is always less than 1, but varies between a certain range with the motion of the SC and the asteroid. It can be seen from the Figure 4.9, α varies from 0° to 360° and from analysis of the Eq. (4.123), that the maximum perturbing accelerations occur at the angles 0° & 180° whereas, the minimum accelerations occur at 90° & 270°, which is intuitive. The scalar potential representing this acceleration can be introduced as,

$$U_P = -G\, m_d\left(\frac{1}{r_{id}} - \frac{\boldsymbol{r}_i\cdot\boldsymbol{r}_d}{r_d^3}\right) \tag{4.125}$$

such that,

$$\boldsymbol{a}_P^{\mathtt{I}} = \nabla U_P \tag{4.126}$$

If there are a number of bodies perturbing the SC, the potentials can be found for each individual body and then summed up to give the total perturbing acceleration (Scheeres (2012)). Since, this is already in the asteroid reference frame the perturbing torque can be given by,

$$\boldsymbol{T}_P^{\mathtt{I}} = \boldsymbol{r}_i \times \boldsymbol{F}_P = \boldsymbol{r}_i \times m_i \sum_{d \neq i,k} \nabla U_P^d = \boldsymbol{r}_i \times m_i \, \boldsymbol{a}_P^{\mathtt{I}} \tag{4.127}$$

Using Eq. (4.124) and comparing perturbing accelerations due to the Sun or other planets, it can be seen, that the ratio m_d / r_d^2 is approximately 10^4 to 10^6 times more for the Sun as compared to any Solar System planets. Therefore, for simplicity the system can be considered as a three body system with just the Sun as a source of the perturbing force. This has been established with respect to the inertial asteroid centered reference frame (`I`) and may need to be converted to different reference frames as per requirement.

5

Fundamentals of Autonomous Guidance

The trajectory planning, sensing and control of the SC is carried out by the onboard GNC system to achieve SC manoeuvres and target pointing, as required by the mission objectives discussed in Section 2.3.3. In the previous chapter, we discussed the kinematics and dynamics of the SC, which dictate how the SC moves in space in response to different forces and disturbances acting on it. The GNC system is responsible for the safe steering of the SC in response to the dynamics and kinematics. Simply put, the navigation subsystem answers *"Where am I?"*, the guidance subsystem *"Where am I going?"* and the control subsystem *"How do I get there?"*. This chapter discusses the basic concepts of GNC systems, how guidance algorithm is dependent on navigation and the approach towards developing the guidance algorithm in the next chapter.

For an asteroid mission, the onboard GNC system needs to deal with a highly dynamic and perturbed environment while meeting autonomy requirements to perform time urgent operations, complex fault responses and stringent pointing requirements. Mapping (Proximity operations) and touch and go (TAG) descent in these low gravity, atmosphereless small Solar System objects are the future mission prospects as discussed in the Chapter 2. A six DOF GNC system, with non-linear trajectory optimisation and adaptive filtering to respond to internal and external dynamic events enables the autonomy of the SC along with the target relative navigation. The requirements for the mission phases of mapping and descent to develop an autonomous guidance system using quaternions and dual quaternions will be discussed in this chapter. The problem formulation for the descent problem will also be presented here and the same for the mapping phase can be found in Appendix A.

5.1. Elements of GNC System

The GNC system as discussed is an advanced system, that enables the SC to perform the required mission safely and precisely. In this section, we discuss the fundamental components of GNC focusing on both orbiting and descent guidance. We will first define the function of the three subsystems and how they interact with each other and formulate a general guidance problem each for mapping and descent phases.

5.1.1. Definitions of Guidance, Navigation and Control

To develop a guidance algorithm, a clear and basic understanding of the three subsystems is necessary. Definitions for them have been adapted from ECSS-E-60C (ESA-ESTEC (2013)) and their interpretation for the mapping and descent phase have been elaborated.

Guidance *function of the system to define the current or future desired state.*
It is a mathematical black box, that calculates and commands the required acceleration profile for the SC to have the required attitude orientation and trajectory as per the mission phase. It enables the SC to achieve the desired end conditions.

Navigation *function of the system to determine the current or future estimated state from measured state.*
The state is measured by means of onboard sensors like horizon or star sensors, cameras, Sun sensors,IMUs (gyroscopes and accelerometers). The estimated state is obtained by means of navigation filters like the extended Kalman filter.

Control *function of the system to derive control commands to match the current or future estimated state with the desired state.*
It is the onboard processing and routing of commands from guidance and navigation to activate actuators like thrusters, reaction wheels, control moment gyroscopes or aerodynamic surfaces for achieving reactive forces on the vehicle. This helps achieve the required spatial alignment and stabilisation of the vehicle.

5.1.2. GNC Architecture and System Interaction

A basic GNC architecture has been shown in the Figure 5.1. This architecture is tailored as per the mission requirements of the SC. The guidance and control form a part of the mission vehicle management (MVM). It is responsible for the appropriate mode selection of the GNC system. Modes here refer to different mission phase requirements, for example *Orbit Control Mode*, *Sun-Keeping Mode*, *Sun Hibernation Mode*, etc. It also bears the responsibility for the safety of the SC by fault identification, detection and recovery (FDIR) and to also put the GNC into its nominal or redundant mode. It, therefore, also selects the GNC hardware, that needs to run in these specific modes.

On an actual SC, the attitude controller is responsible for the attaining the required SC attitude. It utilises the reference state from the guidance and the measured state from the navigation and the difference between the two is converted to commands for the actuators like reaction control wheels or thrusters to execute. But this command has to be within the capabilities of the actuators. There needs to be proper selection of the combination of thrusters. The trust vector can be controlled by using a different combination of thrusters or a gimballed thrusters and the thrust magnitude by throttle control or switching the thrusters on and off.

The navigation system estimates the pose and attitude of the SC by means of navigation instruments and filters. These estimates are then provided in the feedback loop of the GNC system to make the necessary adjustments or corrections in the next iteration or time step. The measurements are prone to noise in the instruments due to various reasons, and the estimate can only be obtained with a certain accuracy. Hence, navigation errors are unavoidable and their impact on the achievable guidance precision and precision in the command execution need to be kept in consideration.

As can be seen, the guidance system provides the command to point the SC in the required direction or to follow a certain trajectory by using the best possible estimate of the current state of the SC. In this fashion, it guides the SC through space and time from the current state to the desired state.

5.1.3. Separating Guidance from Navigation

The discussion in the above subsection brings us to the question if the guidance and control logic can be developed by treating them as a separate entity from navigation. Pfeiffer (1968) discusses the consequences and assumptions for separating the guidance and navigation as presently is the practice of developing guidance algorithms. The navigation system as we discussed is prone to measurement noises and also the real system undergoes disturbances, which cannot be perfectly modelled and hence there is process noise. Even though the measurement and process noise is taken care of in the navigation filters, it is not perfect and as a result,

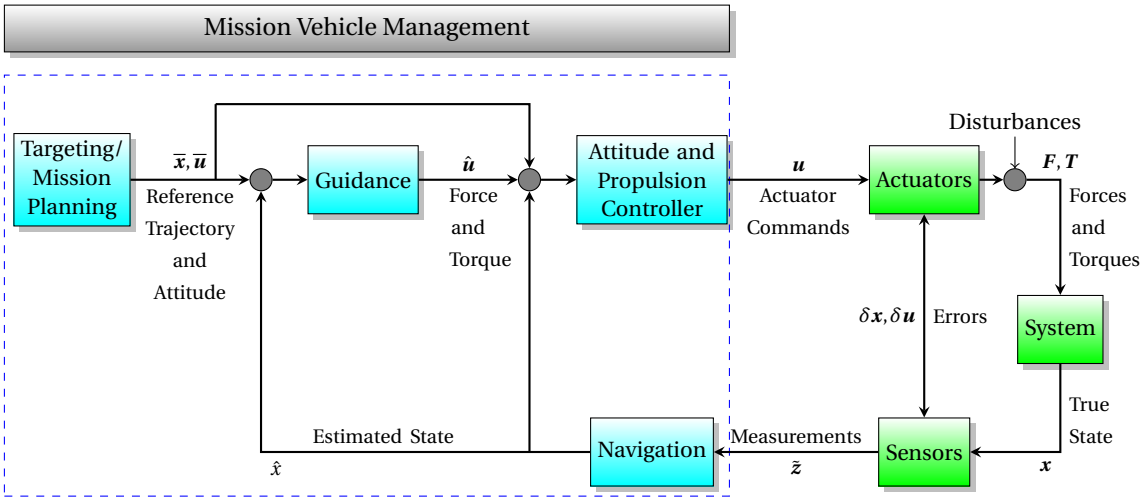


Figure 5.1: GNC System Architecture and Interaction

the state is not known perfectly and cannot simply be defined by the position, velocity and system parameter vectors. The state needs to be defined additionally with statistical moments of the distribution of these parameters, which leads to the problem becoming a stochastic control problem. Developing deterministic guidance for such a problem would, therefore, be incorrect.

To develop a guidance algorithm separately from the navigation as a deterministic optimiser, certain assumptions need to be made. Pfeiffer (1968) discusses, that in case the random navigation errors and random systematic errors are small, stochastic guidance analysis is not required. Also in case, the execution of the guidance is with continuous thrust, stochastic treatment is not required. However, in case of the sequence of small thrust operations at unspecified times we cannot assume the problem to be deterministic. For the descent as well as mapping phase we deal with impulsive thrusts, but at specified times. Hence, for the purpose of this thesis, the guidance problem will be treated as deterministic.

Although we assume the guidance problems to be deterministic, sensitivity analysis should be performed along with navigation. This will help quantify the impact of the precision of the navigation on guidance. Therefore, the cost of assuming the problems to be deterministic can be found out. A benefit of this is, that the guidance algorithms could be tuned to be more robust and less sensitive to navigation errors. This will be left as future work.

5.1.4. General Guidance Problem

In this section, we provide the general 3DOF guidance problem for both the phases of mapping and guidance. In the mapping phase guidance plays the role of designing the optimal trajectory to maximise asteroid observation by the SC and for the descent phase guidance has the task of computing a suitable acceleration profile so the SC can achieve the desired end state or conditions. Hence the descent phase will be posed as a two-point boundary-value problem (TPBVP) whereas the mapping guidance problem as the initial value problem (IVP). The initial problem for descent phase is stated below as Problem 1A and mapping phase as Problem 2A in Appendix A.

➽ PROBLEM 1A: Descent phase general guidance problem (fixed time)

Given: Initial state at time $t = t_0$

$$\boldsymbol{r}(t_0) = \boldsymbol{r}_0, \quad \boldsymbol{v}(t_0) = \boldsymbol{v}_0, \quad m(t_0) = m_0 \tag{5.1}$$

where $\boldsymbol{r}(t)$, $\boldsymbol{v}(t)$ and $\boldsymbol{m}(t)$ are the position, velocity and mass of the SC at time t. The objective is to find an acceleration profile, which in turn results in a trajectory

$$\boldsymbol{a}(t) = \frac{\boldsymbol{T}(t)}{m(t)} + \boldsymbol{g}(t), \quad \forall t \in [t_0, t_f] \tag{5.2}$$

where $\boldsymbol{a}(t)$, $\boldsymbol{T}(t)$ and $m(t)$ are the acceleration, thrust and instantaneous SC mass at time t and t_f is the final time. The final state of the spacecraft is, therefore, given by,

$$\boldsymbol{r}(t_f) = \boldsymbol{r}_f, \quad \boldsymbol{v}(t_f) = \boldsymbol{v}_f, \quad t_f \text{ given} \tag{5.3}$$

The system is defined by some constraints, which could be on the state or control as below,

$$\begin{aligned} \boldsymbol{g} &\leq 0 \\ \boldsymbol{h} &= 0 \end{aligned} \tag{5.4}$$

This problem has been solved using many different methods, but the most promising method so far has been, that of convex optimisation. In this thesis we will work with a novel hybrid approach of an extension of this method -*successive convexification*- and -*dual quaternions*- to analyse the advantages and disadvantages of such a combination for autonomous SC operation in real time.

5.2. Technological Considerations for Autonomous G&C

The orbits of asteroids, their detection and visualisation as well as other physical properties are presently studied with the help of ground-based telescopes or Earth satellites or flyby spacecrafts. After an asteroid has been studied, mission profiles are designed ensuring safety in missions and predetermined trajectories

and mission completion times. This makes the mission dependent on human intervention in case of unforeseen circumstances. Due to the massive distance between small bodies and Earth communication delay in real time becomes a bottleneck in critical situations, as we have seen in the case of the Philae lander for Comet 67P from Rosetta. This has called for the autonomy in future asteroid missions, which begins with relative navigation and guidance. This enables the SC to take decisions in case of unforeseen circumstances for ensuring mission safety.

With the previous sections describing the fundamental concepts of guidance and control, it can be seen, that the system is very complicated and would have specific technological requirements for performing the mission objectives during the mapping and descent phase. In the mapping phase, the general surface topography needs to be generated in order to select landmarks for the next phase of descent. This calls for certain system requirements to obtain a resolution of the surface images, that would suffice for hazard detection. During the TAG sequence, the descent has to be precise for safe sample collection such, that it can thrust itself away from the surface afterwards for a successful ascent. The SC should be enabled to evade hazards in case of undetected hazard in the mapping phase due to limitations on the resolution. We will discuss these concepts in detail in this section.

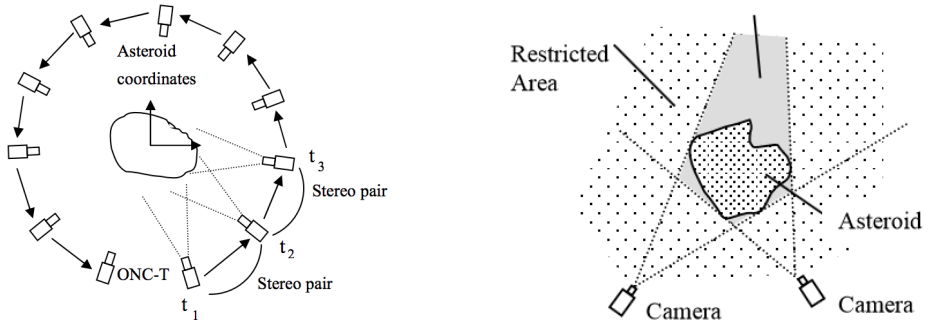
5.2.1. Reasons for High-Resolution Imaging and Precise Ranging

The shape of the asteroid is based on observations from Earth satellites or ground telescopes, it makes the mapping of the asteroid critical to ensure a safe landing location. Hence to ensure a safe and successful descent to the asteroid an accurate estimation of the relative motion between the SC and asteroid and its shape is required. This requires the SC to be capable of navigating relative to the asteroid. This is achieved with the help of navigation sensors like star trackers (STTs), inertial reference units (IRUs), optical navigation cameras (ONCs), lidar and laser range finder (LRF). Our first mission objective is the global mapping of the target asteroid. With the available technologies at present, two methods exist: "optical imaging to estimate the motion and shape of the asteroid (Sawai et al. (2001))" or "shape model construction using laser altimeters (Seabrook et al. (2017))".

Optical Imaging

Motion estimation can be carried out by means of Global Control Points (GCPs). The SC uses the ONCs to estimate the asteroid relative emotion by tracking feature points (GCPs) on the asteroid surface. The coordinates of the GCPs in the asteroid frame can be calculated from the ONC images. Equations that are used to calculate the asteroid motion in terms of the camera coordinates can be found in Maruya et al. (2000).

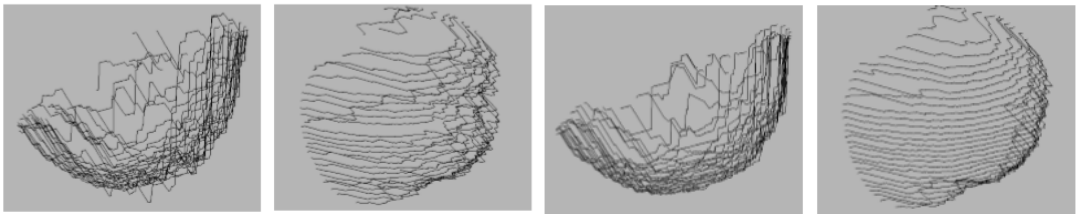
The 3D model is built using a technique called the *motion stereo method* (Sawai et al. (2001)). The ONCs image the asteroid from different viewpoints. The SC maintains a hovering position and uses the asteroid rotation to obtain multi-viewpoint images as shown in Figure 5.2a. Computations for the global mapping are performed on these images using the motion stereo technique, which can be found in detail in Sawai et al. (2001). The accuracy of this method is reliant on the surface texture of the asteroid, which means in the absence of distinctive texture, the matching results are erroneous. To achieve the required accuracy in such situations the principle of shape from limb is used. The asteroid image obtained from the ONCs is divided



(a) ONC positions represented on asteroid coordinates (Maruya et al. (2000)).

(b) Principle of shape from limb (Seabrook et al. (2017)).

Figure 5.2: Optical imaging method to estimate shape and motion of asteroid.



(a) Simulated surface profile using Lambertian reflection (Sawai et al. (2001)).

(b) Simulated surface profile using Hapke reflection (Sawai et al. (2001)).

Figure 5.3: Wireframe representation of matching the asteroid surface profile using reflection models.

into an asteroid region and background region as shown in Figure 5.2b. The 3D object area containing the asteroid is computed from the asteroid region and the restricted area, that holds no intersection with the asteroid from the background region. Since, multi-viewpoint images are taken, circumscribed polygons can be obtained from the 3D object areas. Using circumscribed polygons increases the matching accuracy by reducing the matching errors, that expand the map. The stability of this technique is high, but it fails to prove itself in concave regions.

Another parameter that affects the accuracy of this mapping technique is the asteroid surface reflection. The rotation of the asteroid affects the intensity of the reflected light from its surface, since the incidence and emission angles change with its rotation. This leads to the requirement of modelling the reflection. There are two existing models; Hapke and Lambertian, the latter being less precise. The Lambertian model is dependent on only the incidence angle is given by the equation:

$$I = k\cos(i) \tag{5.5}$$

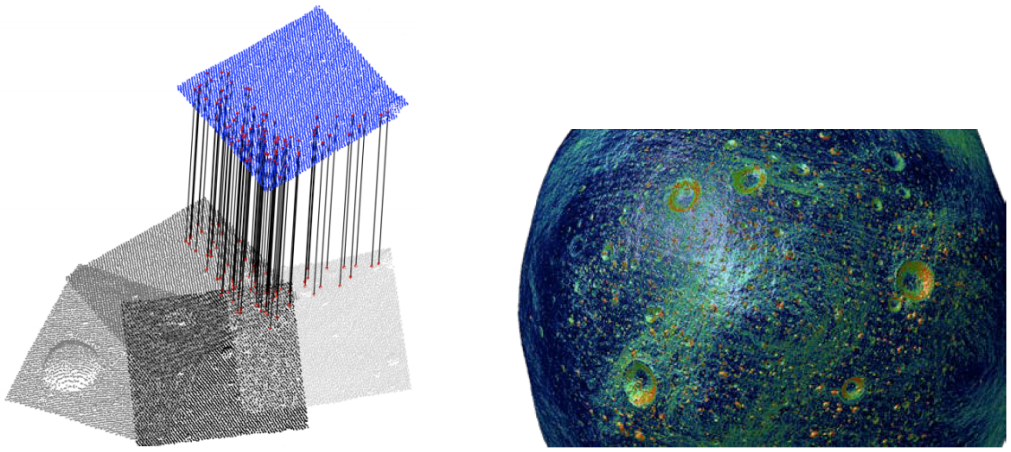
where, I, k and i are the intensity of a point, constant coefficient and incidence angle respectively. The Hapke model as described in Hapke (1986) has two specific characteristics, that make it more accurate, firstly the intensity curve has its peak at the equivalence of the incidence and emission angles called the *opposition effect* and secondly the change in the intensity, which changes in the incidence angle outside the area of the opposition effect is relatively small. The difference in matching accuracy using the two reflection models is shown in Figure 5.3a and 5.3b, and it can be seen that the Hapke illumination model gives more accurate results.

Laser Altimetry

A Laser Ranger (LR) or Light Detection and Ranging (LIDAR) measures the distance to the target by illuminating it with light or laser pulses. This optical remote sensing technique measures the time of flight of the light pulse from the emitter to the target surface and back to the receiver to determine the distance to the target. Laser altimetry is nothing, but the same principle and can be used to construct the topographic map of the asteroid. The navigation lidar is present in conjunction with the laser altimeter used for asteroid mapping, which can be used to provide precision ranges as part of the data input for the navigation solution. The laser altimeter is capable of taking high-resolution raster scans of the surface. While the SC orbits the target small body, overlapping scans across its entire surface are obtained. This high-resolution spatial dataset generates a high fidelity shape model of the target body.

The altimeter can send pulses in the required direction by means of an inbuilt scanning mirror. The origin and direction of these pulses, as well as the mirror scan patterns, can be predetermined, but with autonomous navigation, the MVM can be made capable of commanding them based on the current orbit orientation. The images from the received pulses are a set of overlapping raster scans, that need to be adjusted in a reasonable amount of time to construct the asteroid's shape. For faster computations, key points are generated for the matching method to find common features. These key points are generally surface features, that can be obtained from the point clouds using a mapping tool, which is very beneficial, since they can be used for hazard avoidance later on during the descent phase. As can be said intuitively this reduces the data points to be processed massively for the point cloud registration process.

The first step is to sequentially correct the offsets from the overlapping data using rotational and translational transformations. Using one set of key points as a baseline (usually the first data registration) a key point shape is generated one cloud at a time. Then the set of key points with the largest number of feature



(a) Point cloud registration by OLA using sequential keypoint matching. (Seabrook et al. (2017))

(b) A simulated OLA scan of the asteroid surface. (Dickinson et al. (2012))

Figure 5.4: The technique of achieving a simulated asteroid surface used by OLA, onboard OSIRIS-REx.

matches with the baseline is searched. Least squares are used to minimise the distance between the baseline and the current key point set to transform them rigidly. This can be seen in Figure 5.4a. All the raster scans with these matching key points are exhausted and then the next baseline is chosen. These alignment errors get accumulated and propagated by this sequential method. In order to minimise these errors they are distributed globally and final registration of each data set position is performed using a General Procrustes Analysis[1] (GPA). This results in dataset largely self-consistent while adding the raster scans and building it into a close shaped model. After the errors are minimised the mapping tool is used to construct a grid-shaped model. A simulation by Optical Laser Altimeter (OLA); onboard OSIRIS-REx of the asteroid Bennu is shown in Figure 5.4b.

There can be a discrepancy in the measurement due to attenuation or dispersion of the signal due to reflection off dust and irregular surface of small bodies. These errors can be handled with the reflection models, Hapke or Lambertian. Errors in determining the distance also depend on the wavelength, power, pulse width of the laser pulse and other factors like the secular or non-smooth surfaces, a measurement error of the rotating mirror angle and the clock error. These errors can be mathematically modelled to be taken care of prior to the mission.

5.2.2. Reasons for Hazard Detection

Hazard detection and avoidance (HDA) is key to safely landing an SC on the surface of any target body. In the past missions have been lost to landing in hazardous areas and therefore, it became a need for any space mission, which had the objective of extremely close proximity operations. In case of small body missions, the complexity for the HDA increases, since apriori surface information of the target body is not available and neither is communication from Earth without delay, unlike Lunar landers or Mars landers. It becomes a need to have an onboard algorithm for the SC to be self-aware of the possible hazardous locations and avoid them as sites for landing or descent.

The high-resolution images obtained during the asteroid mapping either by optical or laser imaging play a very important role in HDA. As can be seen from the current state of the art instruments resolution of the surface in the range of centimetres can be obtained. This helps provide information about surface features like slopes, boulders, rocks, craters, regolith areas, etc. in real time. A comparison for the various algorithms for crater recognition, which can be used in the mapping tool can be found in Woicke et al. (2018) and similar algorithms exist for other surface features as well. The SC is, therefore, able to characterise the observed surface well before landing or descent and can choose landing sites while mapping itself. This increases the probability of a safe landing before the close proximity of the SC to the asteroid. Although fail-safe methods need to be accommodated in the automatic HDA for the SC to be able to avoid a hazardous landing site when it is already on its optimised descent path and re-optimise for a new safe landing site.

[1] GPA is an iterative algorithm, that uses reduction in the sum of squared differences between original and adjusted positions.

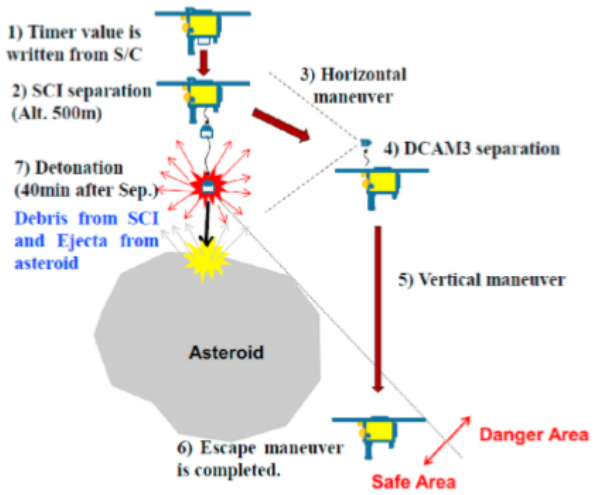


Figure 5.5: Kinetic Impact Operation Sequence of Hayabusa for sample collection. (Tsuda et al. (2017))

5.2.3. Reasons for Precision Descent/Landing

As we have already established, for descent or landing on a small body the autonomous navigational accuracy has to be high. Along with the autonomy for descent/landing, the system must have the flexibility guide the SC to a desired targeted point on the surface with zero velocity or minimal velocity. The most important reason for precision descent is the safety of the SC. Descending to or landing on hazardous surface feature would risk failure of the mission. In the case of a TAG descent, the main objective is sample return. This means the SC has to approach the surface safely whilst also being able to leave safely. Such a mission to asteroids in the NEA region (3 AU) entitles the SC to have larger solar panels (length of 14m for Rosetta). But for the descent phase as discussed in the Section 2.3.6, the SC will have folded solar panels to be similar to the OSIRIS-REx SC. Precision landing is a need to avoid any kind of damage to the SC for return.

As discussed in the scientific objectives of a small body mission, a sample return is needed to study the target bodies internal structure and subsurface materials. This would shed light on the formation history of the asteroid while helping understand the processes in the formation stages of the Solar System. For sample collection from an asteroid, a site that has regolith or loose rocks will not be chosen, since they have undergone weathering due to physical or chemical processes. A site with a conjugated surface is chosen, that has to be loosened by an impactor for the SC to be able to collect samples from deeper regions in the impact area, that are still pristine. This requires the SC to precisely descend to a range above the selected impact site. The impact area is usually in the range of a few metres (Tsuda et al. (2017)) depending on the asteroid size as well as structure. Once the detonation on the target site is achieved and the ejecta has settled or its velocity is low the SC descends to collect the sample. The impact area, therefore, constraints the descent area of the SC thereby calling for precision. The current Hayabusa 2 mission impact and sample collection profile is shown in Figure 5.5

5.2.4. Relevance to the Guidance System

The technological discussions in the prior subsection do not explicitly discuss their relation to the guidance problem and we will try to establish this in this subsection. High-resolution images of the asteroid are reliant on the capability of the optical cameras or laser altimeter and the navigational accuracy similarly precision landing is dependent majorly on navigational accuracy and HDA is dependent on the instruments being used for observation. But these requirements translate to the guidance algorithm in the form of constraints or assumptions, that need to be established before solving the optimal control problems of mapping and descent.

- For high-resolution imaging, if the mission system design opts for optical imaging, constraints occur on the state space of guidance due to lack of visibility in shadow regions whereas for laser altimetry this constraint need not be forced. Also, there has to be a constraint on the altitude of the SC for consistent imaging.

- For generating the image patches or point cloud maps, the vision-based technologies must always be

pointed towards the target body, which requires a constraint on the attitude of the SC.

- For overlapping image patches or raster scans, the angular rate of the SC attitude has to be constrained below a certain value depending on the instrument specifications.
- For HDA, the guidance algorithm of the SC should incorporate a logic for diversion and re-optimisation of trajectory in case the landing site is found to be dangerous. Since, this situation leads to more expense of propellant, the algorithm must also be very efficient.
- For precision descent, the target site should always be in the field of view of the SC, which again constrains the attitude while descent.
- Another requirement for precision descent is, that the propellant expulsion actuator needs to be gimballed or the SC attitude has to be changed to obtain the required velocities at required distances from the target body.

As we can see, the high-resolution imaging, HDA and precision descent directly impact the guidance making it complex and dependent on many computational resources. The developed autonomous guidance should be able to handle these complexities in combination whilst being efficient.

5.3. Touch and Go Descent Phase

In the descent phase the guidance subsystem has to generate a trajectory from the last position of the mapping phase to the selected landing site. It also needs to maintain its attitude within certain bounds to make sure, that the target body is in sight. The HDA system is still active in this phase to make sure in case of contingency, that a retargeting is initiated for the guidance algorithm to reevaluate for a new optimal trajectory. It would generate the required acceleration profiles for this and provide it to the control subsystem to execute them. While generating the acceleration profiles it also needs to take care of the control actuator limitations.

The TAG descent requires a combination of long ascent thrusts while preserving a desired safe attitude to prevent contact with the surface of the target body. These require a tight coupling between the trajectory and attitude control and a six DOF guidance and control (3 for the position, 3 for the attitude) (Xinfu et al. (2017)). As discussed in the optimisation algorithm heritage, we would approach this problem with *successive convexification*. Advantages of this method are, that it guarantees convergence for a well-posed convex problem, the solution is the global optimum and is a solution to the original problem, a number of efficient solvers have been developed for this kind of problem and constraints and penalties can be imposed. The novel method of this thesis is the combination of the descent problem in the form of a convex problem using dual quaternions with relative states.

5.3.1. Basic Descent GNC framework

The basic architecture for the descent is presented in Figure 5.6. As shown in the flow, the detailed survey during global mapping provides with the hazard map and is fed to the HDA system. The onboard instruments enable navigation and together with the hazard map, the simplified guidance provides a fuel map. The fuel map and hazard map are used by the MVM to make the final landing site selection. The HDA is still active after the final landing site selection and retargeting is still possible. The guidance then generates the optimal propellant consumption trajectory from the initial location to the final location. From this, the control inputs are provided to the thrusters. The true state of the SC is estimated by the navigation.

5.3.2. Refined Problem Formulation

The descent problem is reformulated using the mission scenario. The guidance problem for precision power descent has been phrased by Açıkmeşe and Ploen (2007) as below:

The powered descent guidance problem for pin-point landing is defined as finding the fuel optimal trajectory, that takes a lander with a given initial state (position and velocity) to a prescribed final state in a uniform gravity-field, with magnitude constraints on the available net thrust and various other constraints.

The definition is for a 3DOF problem, but holds true for 6 DOF as well. The "uniform gravity-field" refers to a constant gravitational acceleration or a flat-body model, but in our case we will replace it with the variable gravity field in the problem formulation in Chapter 6.

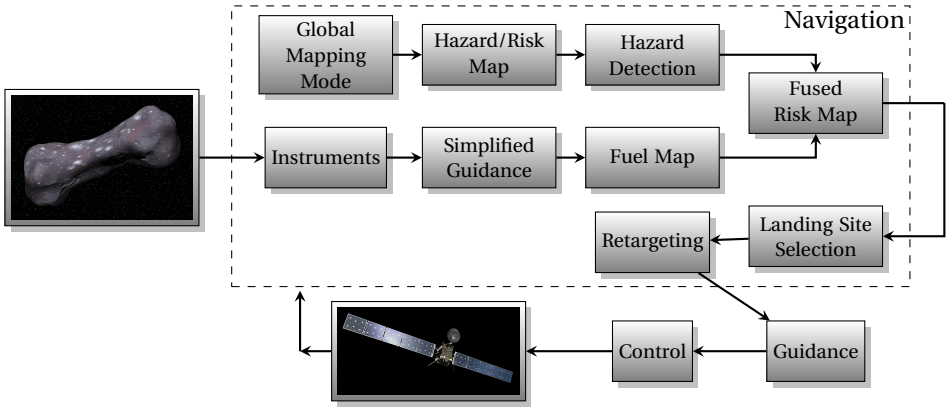


Figure 5.6: Top level GNC architecture for descent to the small body

➽ PROBLEM 1B: Refined descent phase 6-DOF guidance optimisation problem

minimise -$m(t_f)$

subject to:

Kinematics:
$\dot{\boldsymbol{r}}^{\mathtt{I}}(t) = \boldsymbol{v}^{\mathtt{I}}(t), \quad \dot{\boldsymbol{q}}_{\mathtt{B/I}}(t) = \frac{1}{2}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \otimes \boldsymbol{q}_{\mathtt{B/I}}(t)$

Dynamics:
$\dot{m}(t) = -\alpha \| \boldsymbol{T}^{\mathtt{B}}(t) \|, \quad \dot{\boldsymbol{v}}^{\mathtt{I}}(t) = \frac{\boldsymbol{T}^{I}(t)}{m(t)} + \boldsymbol{g}_{\mathtt{I}}, \quad \dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/I}}(t) = \boldsymbol{J}^{-1}\left(\boldsymbol{r}_e(t) \times \boldsymbol{T}^{\mathtt{B}}(t) - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \times \boldsymbol{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t)\right)$

Control Constraints:
$T_{min} \leq \| \boldsymbol{T}^{\mathtt{B}}(t) \| \leq T_{max}$

Boundary Conditions:
$m(0) = m_{wet}, \quad \boldsymbol{r}^{\mathtt{I}}(0) = \boldsymbol{r}^{\mathtt{I}}_0, \quad \boldsymbol{r}^{\mathtt{I}}(t_f) = \boldsymbol{r}^{\mathtt{I}}_f, \quad \boldsymbol{v}^{\mathtt{I}}(0) = \boldsymbol{v}^{\mathtt{I}}_0, \quad \boldsymbol{v}^{\mathtt{I}}(t_f) = \boldsymbol{v}^{\mathtt{I}}_f$
$\boldsymbol{q}_{\mathtt{B/I}}(0) = \boldsymbol{q}_{\mathtt{B/I}_0}, \quad \boldsymbol{q}_{\mathtt{B/I}}(t_f) = \boldsymbol{q}_{\mathtt{B/I}_0}, \quad \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(0) = \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}_0}, \quad \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t_f) = \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}_f}$

Attitude Constraints:
$\zeta_1 \leq \iota \leq \zeta_2$

where, $\dot{m}(t)$ is the mass flow, α is the thrust-to-mass flow conversion factor, T_{min} and T_{max} are the lower and upper bounds on the magnitude of thrust, ζ_1 and ζ_2 are real numbers and ι is the viewing angle between the LOS of the camera and landing site.

As can be seen the problem in 1B is in the inertial frame and this is appealing due to the simpler EOMs. However, this leads to the final time boundary conditions being dependent on time due to the rotation of the target body about itself. This could render the problem intractable. Also, additional constraints of the extended guidance problem (to be discussed in Section 6.4) need to be defined with respect to the landing site, which are convex by nature. To avoid these problems we define the whole problem in the asteroid rotating reference frame. With the formulation in relative frame, constant initial and final boundary conditions can be implemented retaining the tractability of the problem. Chapter 6, provides a step by step explanation through its sections, to the final problem formulation to be used for developing the guidance algorithm for descent.

6

Successive Convex Optimisation

Mathematical optimisation finds the best solution (maxima or minima) out of feasible solutions for a given problem. The standard formulation of an optimisation problem is as follows (Boyd and Vandenberghe (2010)),

$$\begin{aligned} &\text{minimise} && f_0(x) \\ &\text{subject to} && g_j(x) \leq b_j, \quad j = 1,\dots,m. \\ &&& h_l(x) = c_l, \quad l = 1,\dots,p. \end{aligned} \tag{6.1}$$

where, $x = (x_1,\dots,x_n)$ is the *optimisation variable* of the problem, $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective* function of the variable, which needs to be optimised (minimised in the case above). The objective function is subject to the functions $g_i : \mathbb{R}^n \to \mathbb{R}$ and $h_i : \mathbb{R}^n \to \mathbb{R}$ are the *inequality* and *equality* constraints and the constants $b_1,\dots,b_m$ and $c_j,\dots,c_p$ are the bounds on them. The solution of this problem is given by a vector $\boldsymbol{x}^*$, which gives the minimal value of the objective function whilst satisfying all the constraint equations, such that,

$$f_0(x^*) \leq f_0(z) \quad \forall\ z \in \mathbb{R}^n \quad \text{with} \quad g_j(z) \leq b_j \quad \& \quad h_l(z) \leq c_l \tag{6.2}$$

The optimisation problem is classified as *linear* or *non-linear* depending on the form of the objective and constraint functions. A linear program has objective and constraint functions, that satisfy,

$$g_j(\alpha x + \beta y) = \alpha g_j(x) + \beta g_j(y) \quad \forall\ x, y \in \mathbb{R}^n \quad \alpha, \beta \in \mathbb{R} \tag{6.3}$$

The small Solar System body descent problem is plagued with non-convex constraints as well as non-linearities due to time-varying gravity fields and other perturbing forces. We can, therefore, no longer apply methods to solve linear problems directly. This chapter walks us through the process of convex optimisation and how successively convexifying non-linear or non-convex problems can utilise it.

6.1. Basic Concepts of Convex optimisation

Convex optimisation is a rather young development as compared to well-established methods like least squares or linear programming and was explored to the current status mostly in the 20^{th} century. In this short period, it has proved its worth with numerous beneficial properties. Açıkmeşe and Ploen (2007) presented their work to find a soft-landing guidance algorithm for a powered descent on Mars implementing convex optimisation. Since then there have been quite a few developments in the treatment of the powered descent problem as discussed in Chapter 2. We will discuss some basic concepts in this section and work our way towards understanding the presently developed algorithms and finally formulate the problem using successive convexification using dual quaternions.

6.1.1. Convex Sets and Functions

A convex optimisation problem calls for convex objective and constraint functions, which means they should be of the form,

$$g_j(\alpha x + \beta y) \leq \alpha g_j(x) + \beta g_j(y) \quad \forall\ x, y \in \mathbb{R}^n \quad \alpha, \beta \in \mathbb{R} \quad \alpha + \beta = 1, \quad \alpha \geq 0, \beta \geq 0 \tag{6.4}$$

A function is, therefore, convex if the function of the weighted average is less than or equal to the weighted average of the function. As can be seen from the form of convex objective or constraints, convex problems are nothing but a generalisation of linear problems by replacing the equality by inequality and with a restriction on α and β.

Convex sets are fundamental to understanding the theory of convex optimisation and hence will be discussed briefly. First we define a line segment between two points $x_1,\ x_2 \in \mathbb{R}^n$ and $x_1 \neq x_2$ as,

$$y = \theta x_1 + (1-\theta)x_2, \quad \theta \in \mathbb{R}, \quad 0 \leq \theta \leq 1 \tag{6.5}$$

An *affine set* is, such that a line through any two points in the set lies in the set itself, meaning that a set, $\mathscr{C} \subseteq \mathbb{R}^n$ is an *affine set* if for any $x_1, x_2 \in \mathscr{C}$ and $\theta \in \mathbb{R}$, its true, that $\theta x_1 + (1-\theta)x_2 \in \mathscr{C}$, An affine combination, $\theta_1 x_1 + \ldots + \theta_k x_k$, with $\theta_1 + \ldots + \theta_k$ lies in $\mathscr{C}$ too. *Convex sets* are in a way extension of affine sets where θ is restricted between 0 and 1 along with $\sum \theta_i = 1$. The set $\mathscr{C} \subseteq \mathbb{R}^n$ is convex for any $x_1,\ x_2 \in \mathscr{C}$ and $0 \leq \theta \leq 1$, such that $\theta x_1 + (1-\theta)x_2 \in \mathscr{C}$. "If every point in the set can be seen by every other point in the set, along an unobstructed straight path, its a convex set" (Boyd and Vandenberghe (2010)). Figure 6.1 represents geometric examples of convex and non-convex sets. A convex combination can, therefore, be represented as,

$$\theta_1 x_1 + \ldots + \theta_n x_n \in \mathscr{C}, \quad x_k \in \mathscr{C}, \quad \theta_i \geq 0, \quad \sum \theta_k = 1,\ n = 1, \ldots, k \tag{6.6}$$

A special kind of convex set is the convex cone, which is used in the formulation of *second order conic programming*, which is a method to treat convex optimisation problems. The set $\mathscr{C}$ is a convex cone if for any $x_1, x_2 \in \mathscr{C}$ and $\theta_1, \theta_2 \geq 0$, we have,

$$\theta_1 x_1 + \theta_2 x_2 \in \mathscr{C} \tag{6.7}$$

Geometrically the above equation represents a 2D slice of pie, with its apex at 0 and its edges passing through x_1 and x_2. Figure 6.2 represents this convex cone on the left and also shows an example of a cone, that is non-convex on the right, since the line passing through x_1 and x_2 does not lie within the set. With this the second-order cone or the *Euclidean*[1] norm cone in can be represented as,

$$\begin{aligned} \mathscr{C} &= \{(x,t) \mid \|x\|_2 \leq t,\} \subseteq \mathbb{R}^{n+1} \\ &= \left\{ \begin{pmatrix} x \\ t \end{pmatrix} \middle| \begin{pmatrix} x \\ t \end{pmatrix}^T \begin{bmatrix} \boldsymbol{I}_n & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} x \\ t \end{pmatrix} \leq 0,\ t \geq 0 \right\} \end{aligned} \tag{6.8}$$

The second-order cone in $\mathbb{R}^3$ is also known as *quadratic cone* or *Lorentz cone* or *ice-cream cone* and can be represented as $\mathscr{C} = \{(x_1, x_2, t) \mid \|x_k\|_2 \leq t,\ k = 1,2\} \subseteq \mathbb{R}^3$ and is shown in the Figure 6.3.

To formulate an optimisation problem as a convex one, all the functions in the problem must be convex. Problems like the descent guidance do not have convex constraints due to the lower bound on the thrust and also the dynamics governing the motion of the SC is non-linear. Hence, it is necessary to be able to convert these to convex functions to be able to solve the problem as a convex optimisation problem. A function, $f : \mathbb{R}^n \to \mathbb{R}$ is said to be convex if it satisfies the following three conditions:

(i) dom f is a convex set

(ii) All $x,\ y \in \text{dom} f$

(iii) $0 \leq \theta \leq 1$

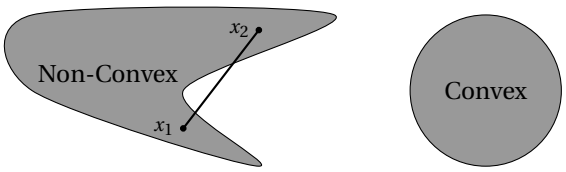
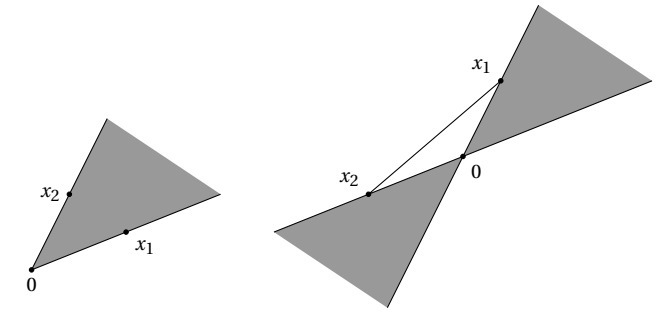


Figure 6.1: Convex and non-convex set examples in 2D.

[1] Euclidean norm is the second norm given by $\|x_1 - x_2\|_2 = \sqrt{(x_1 - x_2)^T (x_1 - x_2)}$



(a) $\theta_1, \theta_2 \in \mathbb{R}_+$ (Convex) (b) $\theta_1, \theta_2 \in \mathbb{R}$ (Non-Convex)

Figure 6.2: Convex and non-convex cone examples in 2D.

This leads to the inequality $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$ to hold true. Figure 6.4a shows the geometric interpretation of this inequality. As can be seen, that for a function to be convex, the line joining any two points of the function in the set must lie above the graph of the function of all the points. Since, it cannot be intuitively said if a function is convex or not by simply looking at the inequality function, the way to recognise the function's convexity is, that it must fulfill the first and second order necessary conditions.

First Order: If f is differentiable, then f is convex iff dom f is convex and

$$f(x) \geq f(x_0) + \nabla f(x_0)^T (x - x_0), \quad \forall x, y \in \mathrm{dom} f \tag{6.9}$$

Second Order: If f is twice differentiable, then f is convex iff dom f is convex and $\nabla^2 f$ is positive semidefinite for all $x \in f$

$$\nabla^2 f \succeq 0 \tag{6.10}$$

As can be seen the right hand side of the first order condition is a Taylor approximation of the function $f(x)$ and the inequality makes it a global underestimator of the function if its convex, i.e., the line shown in Figure 6.4b always lies below the graph if the function is convex. This enables deriving global information from local information making it the most important property of convex functions and resulting in beneficial properties in convex optimisation problems Boyd and Vandenberghe (2010). The second order condition is, nothing but that the Hessian must be positive semidefinite, i.e., the derivative must be non-decreasing and the function of x should have an upward curvature.

These conditions hold for differentiable functions, but non differentiable convex functions exist, eg. $f(x) = |x|$. As can be interpreted at $x = 0$, there is a kink in the graph and is a non differential at the point, but it is a convex function. hence the above conditions are not a general requirement for a function to be convex, but are a check for differentiable functions.

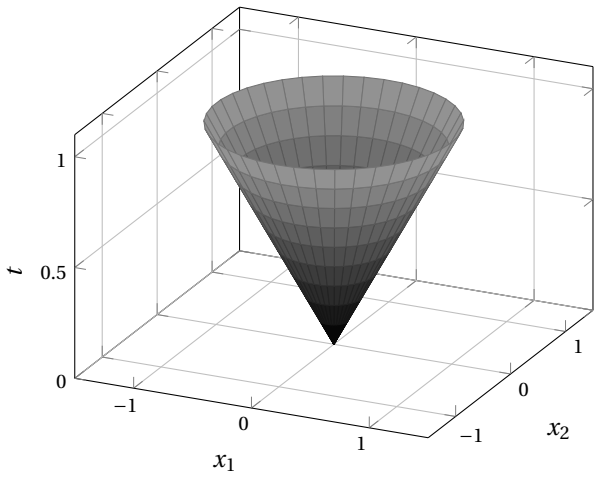
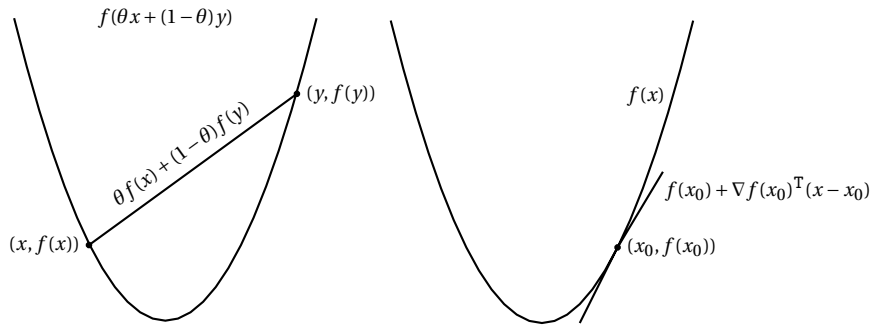


Figure 6.3: Second order cone boundary. It represents a filled cone.



(a) The line joining the two points will always lie above the graph

(b) First order condition for convex functions

Figure 6.4: Graphical representation and properties of a convex function.

6.1.2. Convex optimisation Problems: Second-Order Cone Programming

Convex optimisation problems possess certain properties, that enable efficient solving of the problem with robust solutions. The major difficulty is the recognition and formulation of the convex problem. Once the two are achieved, solving the problem is almost as similar as least squares or linear programming. Here we will discuss SOCP in detail, since it is the form used for the descent guidance. We put forth the form of *Convex optimisation problems* (Boyd and Vandenberghe (2010)),

$$\begin{aligned} &\text{minimise} && f_0(x) \\ &\text{subject to} && g_j(x) \leq 0, \quad j = 1, \ldots, m. \\ &&& a_l^T x = b_l, \quad l = 1, \ldots, p. \end{aligned} \tag{6.11}$$

In comparison with the general optimisation problem, the convex optimisation problem has the following three requirements,

(i) The objective function f_0 has to be a *convex* function; and

(ii) The inequality constraints $g_1, \ldots, g_m$ have to be *convex*; and

(iii) The equality constraints $a_l^T x - b_l$ or $h_1, \ldots, h_p$ must be *affine*.

The advantage of the constraints being convex is the feasible set of the problem from the intersection of all the constraints is also convex. This is, because of the property, that intersection of convex sets is convex (Boyd and Vandenberghe (2010)).

$$\mathscr{D} = \bigcap_{i=0}^{m} \text{dom } g_j \tag{6.12}$$

For the problem to be categorised as a SOCP, the following requirements must be met,

(i) The objective function f_0 has to be a *affine* function; and

(ii) The inequality constraints $g_1, \ldots, g_m$ have to be *convex cones*; and

(iii) The equality constraints $a_l^T x - b_l$ or $h_1, \ldots, h_p$ must be *affine*.

The form of the convex problem is updated to SOCP as,

$$\begin{aligned} &\text{minimise} && f^T x \\ &\text{subject to} && \|A_j(x) + b_j\| \leq c_j^T x + d_j, \quad j = 1, \ldots, m. \\ &&& Fx = g \qquad\qquad Hx \leq s \end{aligned} \tag{6.13}$$

where $A \in \mathbb{R}^{k \times n}$ and $F \in \mathbb{R}^{n+1}$. Comparing with Eq. (6.8), it can be seen, that the inequality constraints represent a second-order cone.

$$\mathscr{C} = \{(A_j x + b_j, c_j^T x + d_j) \mid \|A_j x = b_j\| \leq c_j^T x + d_j,\ j+1, \ldots, m\} \subseteq \mathbb{R}^{k+1} \tag{6.14}$$

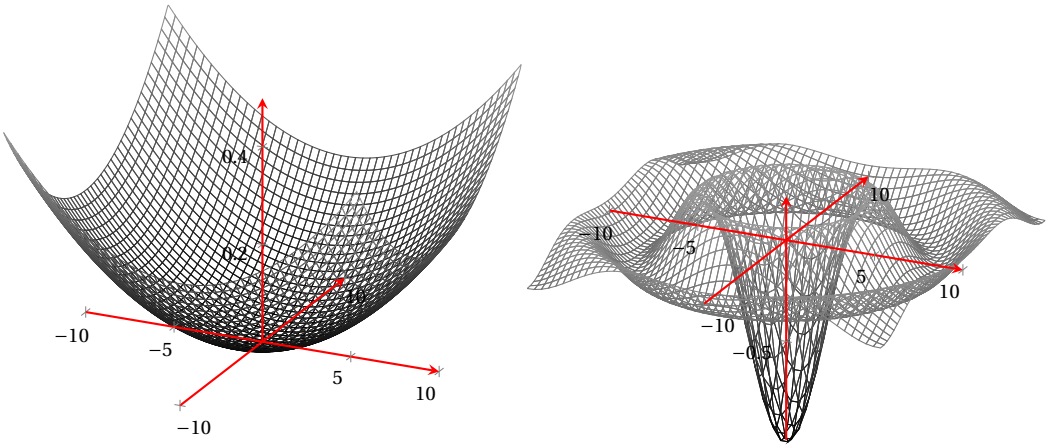


Figure 6.5: Representations of a convex and non-convex function in 3D.

The obvious reason for using convex optimisation is the necessary existence of *only one global minimum* for convex function over its domain. The probability of getting stuck at a local minimum is, therefore, nil. Also, the negative gradient of the function will always lead to the optimum. A parabola is used to represent a convex function in Figure 6.5a, since it is easy to visualise, but it does not generalise convexity. In reality, convex sets are not intuitive or easy to visualise.

As mentioned earlier the difficult part is the problem formulation in the SOCP form. SOCP possesses remarkable properties, which enable speedy solving of the problem. These solvers are on the verge of being a mature technology. At the end of this chapter, the descent guidance will be formulated as an SOCP.

The next sections are towards understanding the steps how a nonconvex, non-linear problem can be reformulated to a convex one without compromising on the constraints or objective function. The convex guidance developed by Açıkmeşe and Ploen (2007) called *lossless convexification* will be briefly discussed to give an overview of the existing methods. Then we will move on to *successive convexification*, which has been developed and used in a few recent works Mao et al. (2016), Szmuk et al. (2017), Szmuk and Açıkmeşe (2018), Açıkmeşe et al. (2013) and Xinfu and Ping (2014).

6.2. Lossless Convexification

Referring back to Problem 1B, it can be seen, that the source of non-linearity comes from the dynamics due to the non-linear gravitational acceleration of the target body and nonconvexity from the control constraint. The method of lossless convexification by Açıkmeşe and Ploen (2007) provides a way to mathematically manipulate nonconvex functions to a second order cone form, thereby making it convex and enabling the computation of the global minimum to a given accuracy with a deterministic upper bound on the number of iterations for convergence. We discuss each step of the procedure in the section and reformulate the nonconvex constraints to a convex form.

6.2.1. Introduction of Slack Variables

The control constraint dealing with the thrust magnitude given in the optimisation problem is the primary nonconvex constraint in the problem. Once the thrusters have been initiated, they cannot be turned off during the manoeuvre, and there is a minimum level of thrust, at which the propulsion system can operate. Below this level, the operation is unreliable, and hence the thrust magnitude needs to be bounded at the lower end.

$$0 < T_{min} \leq \|\boldsymbol{T}(t)\| \leq T_{max} \tag{6.15}$$

The lower bound defines the nonconvex feasible control space.[2] This can be proved geometrically by considering a two dimensional thrust as follows.[3]

[2] An upper bounded constraint, $0 \leq \|\boldsymbol{T}_{\mathrm{B}}(t)\| \leq T_{max}$ would be convex. Visualising Figure 6.6, with just an upper bound, the set would be a closed circle, which is convex.

[3] For $T_z \neq 0$, the region could be imagined as a partially hollow sphere.

$$sup\{\|\boldsymbol{T}\| \mid T_z = 0,\ \|\boldsymbol{T}(t)\| \leq T_{max}\} = \{\|\boldsymbol{T}\| \mid \sqrt{T_x^2 + T_y^2} = T_{max}\} \tag{6.16}$$

$$inf\{\|\boldsymbol{T}\| \mid T_z = 0,\ 0 < T_{min} \leq \|\boldsymbol{T}(t)\|\} = \{\|\boldsymbol{T}\| \mid \sqrt{T_x^2 + T_y^2} = T_{min}\} \tag{6.17}$$

Considering T_{max} and T_{min}, the constraint is geometrically shown in Figure 6.6 as an annular diagram. Repeating the definition of a convex function, it states, that a line drawn between two points in the convex set must also lie in the set. As can be seen, that the lines passing through the central region of the annulus to connect two points do not lie inside the convex set themselves hence proving the nonconvexity of the function.

Lossless convexification is a simple, but effective way of manipulating the nonconvex function to make it convex. A scalar slack variable is introduced to relax the control constraint, such that the control space becomes *affine*, which makes it inherently convex (Açıkmeşe et al. (2013)). For the thrust control constraint, its magnitude is replaced by the scalar variable, Γ and the variable itself is constrained by the thrust magnitude.

$$0 < T_{min} \leq \Gamma \leq T_{max} \tag{6.18}$$

$$\|\boldsymbol{T}(t)\| \leq \Gamma \tag{6.19}$$

The thrust magnitude has been relaxed by Γ without a lower bound on it, which means that $\|\boldsymbol{T}(t)\| \leq T_{min}$ is feasible in the new problem, but not in the original problem, with the bounded thrust magnitude. The original control space is, therefore, a subset of the relaxed control space, represented as below,

$$\{\boldsymbol{T} \mid 0 \leq T_{min} \leq \|\boldsymbol{T}(t)\| \leq T_{max}\} \subset \{\boldsymbol{T},\ \Gamma \mid 0 < T_{min} \leq \Gamma \leq T_{max}\} \tag{6.20}$$

However, Açıkmeşe and Ploen (2007) have proved, that an optimal solution to the new problem with relaxed constraints will satisfy the original thrust constraints as well. The proof is quite long and hence will not be presented in this thesis and the interested reader can refer to Açıkmeşe and Ploen (2007) for the same. The central conclusion from the proof is, that the optimal control history for both Γ and $\|\boldsymbol{T}\|$ is same as shown below, which also requires $\|\boldsymbol{T}^*\|$ to satisfy the Eq. (6.18).

$$\Gamma^*(t) = \|\boldsymbol{T}^*(t)\| \tag{6.21}$$

Figure 6.7 represents the visualisation of the relaxed control space for understanding the convexity of the new constraints. The combination of both the relaxed constraints leads to a truncated cone as the control space, which is convex by nature. The affine control space representation shown on the axis Γ is represented similarly in the combined figure. The binding values T_{min} and T_{max} form the end caps of the truncated cone at their locations on the Γ axis parallel to the plane of the torque axes.

In Subsection 6.1.2 the required form for a constraint has been put forth in Eq. (6.13). To take advantage of the beneficial properties of SOCPs, the new constraints need to be in the same form. Therefore, any other constraints, that might be needed for the powered descent guidance will be added to the problem in later stages in this form. For now, the newly formulated convex problem is as shown in Problem 1C.

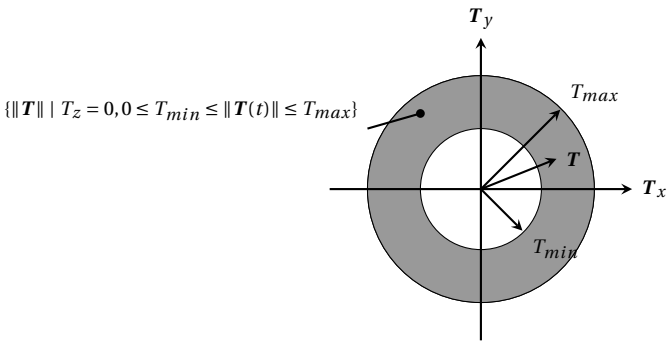


Figure 6.6: Physical representation of the nonconvex control space from the original thrust constraints. The lines passing through the central region have points, that do not lie in the solution set.

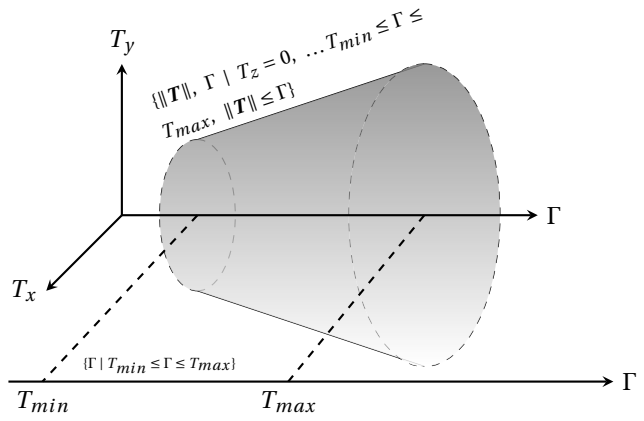


Figure 6.7: Physical representation of the convexified control space using Γ, the slack variable.

➽ PROBLEM 1C: Convexified Control Constraints

minimise $-m(t_f)$

subject to:

Kinematics:
$\dot{\boldsymbol{r}}^{\mathtt{I}}(t) = \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}}(t), \qquad \dot{\boldsymbol{q}}_{\mathtt{B/I}}(t) = \frac{1}{2}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \otimes \boldsymbol{q}_{\mathtt{B/I}}(t)$

Dynamics:
$\dot{m}(t) = -\alpha\Gamma(t), \qquad \dot{\boldsymbol{v}}^{\mathtt{I}}_{\mathtt{B/I}}(t) = \frac{\boldsymbol{T}^{\mathtt{I}}(t)}{m(t)} + \boldsymbol{g}_{\mathtt{I}}$
$\dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/I}}(t) = \boldsymbol{J}^{-1}(\boldsymbol{r}_e(t) \times \boldsymbol{T}^{\mathtt{B}}(t) - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \times \boldsymbol{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t))$

Control Constraints:
$0 < T_{min} \leq \Gamma \leq T_{max}$
$\|\boldsymbol{T}^{\mathtt{I}}(t)\| \leq \Gamma$

Boundary Conditions:
$m(0) = m_{wet}$
$\boldsymbol{r}^{\mathtt{I}}(0) = \boldsymbol{r}^{\mathtt{I}}_0, \quad \boldsymbol{r}^{\mathtt{I}}(t_f) = \boldsymbol{r}^{\mathtt{I}}_f, \quad \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}}(0) = \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}_0}, \quad \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}}(t_f) = \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}_f}$
$\boldsymbol{q}_{\mathtt{B/I}}(0) = \boldsymbol{q}_{\mathtt{B/I}_0}, \quad \boldsymbol{q}_{\mathtt{B/I}}(t_f) = \boldsymbol{q}_{\mathtt{B/I}_f}, \quad \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(0) = \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}_0}, \quad \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t_f) = \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}_f}$

Additional Constraints:
$\|A_j(\boldsymbol{x}) + b_j\| \leq c_j^T\boldsymbol{x} + d_j, \quad j = 1, \ldots, m$

Note: Since, we deal with variable SC mass, the inertia changes as well and the way to deal with this has been explored in Section 7.2.2.

6.2.2. Change of Variables

The Problem 1C has convexified control and additional constraints, but is still not fit for convex solvers due to the non-linearity in the dynamics. The non-linearity in the mass flow dynamics will be dealt with the change of variables. The control variables $\boldsymbol{T}(t)$ and $\Gamma(t)$ will be changed to establish linear second order differential equation, that can ultimately represent a linear, time-invariant, state-space system (LTI). The control variables are transformed to accelerations from thrust forces as below,

$$\sigma(t) \triangleq \frac{\Gamma(t)}{m(t)}, \quad \boldsymbol{\tau}(t) \triangleq \frac{\boldsymbol{T}(t)}{m(t)} \tag{6.22}$$

It is important to mention here, that this change of variables maintains the equivalence in the optimal controls as it was for $\boldsymbol{T}(t)$ and $\Gamma(t)$, i.e., $\|\boldsymbol{\tau}^*(t)\| = \sigma^*(t)$. Using the new variables $\sigma(t)$ and $\boldsymbol{\tau}(t)$ the EOMs for

translational motion and mass flow can be rephrased as,

$$\boldsymbol{v}(t) = \boldsymbol{\tau}(t) + \boldsymbol{g} \tag{6.23}$$

$$\frac{\dot{m}(t)}{m(t)} = -\alpha\sigma(t) \tag{6.24}$$

As can be seen the EOMs are still not linearised due to the non-linear gravity field in Eq. (6.23) and the time dependent mass term in Eq. (6.24). Further steps need to be taken in order to linearise them as well as the other non-linear and dynamics constraints. The cost function and the control constraints need to be changed with these variables. Eq. (6.24) is a first order, ordinary differential equation (ODE). This ODE can be solved as (Boyce and DiPrima (2008)),

$$m(t) = m_0 \exp\left(-\alpha \int_0^{t_f} \sigma(t)dt\right) \tag{6.25}$$

From the above equation, it can be seen, that minimising the exponential function would maximise $m(t)$. Hence, the cost function is reformulated to a new, but valid cost function as shown below,

$$\text{minimise} - m(t_f) = \text{minimise} \int_0^{t_f} \sigma(t)dt \tag{6.26}$$

The control constraints with the new variables are given as,

$$\|\boldsymbol{\tau}(t)\| \leq \sigma(t) \tag{6.27}$$

$$\frac{T_{min}}{m(t)} \leq \sigma(t) \leq \frac{T_{max}}{m(t)} \quad \forall\ t \in [0, t_f] \tag{6.28}$$

But as can be seen the second constraint is again a source of non-linearity due to the time dependent mass. This nonconvexity can be removed by another change of variables, given as,

$$z(t) \triangleq \ln m(t) \tag{6.29}$$

The ODE for mass flow in Eq. (6.25) is changed to,

$$\dot{z}(t) = -\alpha\sigma(t) \tag{6.30}$$

The inequality control constraint is, therefore, now rewritten as,

$$T_{min}\, e^{-z(t)} \leq \sigma(t) \leq T_{max}\, e^{-z(t)} \tag{6.31}$$

This constraint is still not convexified as can be understood from the similar graph of the function $1/z$ and also the exponential cannot be a part of the SOCP constraint. In order to take care of this a Taylor series approximation is used for the the exponential function as below (Boyce and DiPrima (2008)),

$$f(z) = \sum_{n=0}^{\infty} \frac{f^{(n)} z_0}{n!} (z - z_0)^n \tag{6.32}$$

where, z_0 and n denote the nominal, around which the series is to be extended and the n^{th} order derivative respectively. Considering up to the 2$^{\text{nd}}$ order derivative would suffice, since it provides enough precision and also it would keep the lower bound thrust convex, which is the purpose of the variable changes. The lower bound on the thrust can be represented as,

$$T_{min}\, e^{-z_1(t)}[1 - (z(t) - z_1(t) + \frac{1}{2}z(t) - z_1(t))^2] \leq \sigma(t) \tag{6.33}$$

where, z_1 is the nominal point. For the upper bound on the thrust as discussed above, anything above the first order would maintain nonconvexity and therefore, just the first order Taylor expansion will be used.

$$\sigma(t) \leq T_{max}\, e^{-z_2(t)}[1 - (z(t) - z_2(t))] \tag{6.34}$$

where, z_2 is the nominal point. The pre-factors multiplied to the expansion series can be abbreviated as,

$$\mu_1(t) = T_{min}\, e^{-z_1(t)}, \quad \mu_2(t) = T_{max}\, e^{-z_2(t)} \tag{6.35}$$

It is important, that the mass equivalent variables z_1 and z_2 or the nominal points are logical to ensure sufficient accuracy of the newly defined constraints. This can be done by reiterating the definition of z and setting the slack variable Γ to the lower and upper thrust values for the lower and upper bounds respectively.

$$z_1(t) = \ln m(t) = \ln(m_0 - \alpha T_{min} t) \tag{6.36}$$

$$z_2(t) = \ln m(t) = \ln(m_0 - \alpha T_{max} t) \tag{6.37}$$

An additional constraint needs to be put on z, such that the physical bounds on it remain intact.

$$z_1(t) \leq z(t) \leq z_2(t) \tag{6.38}$$

Using all the newly defined constraints the SOCP can be set up for the next phase where we need to convexify the remaining non-linear dynamics. Also using the equations for relative frame kinematics and dynamics we can update the convexified problem in 1C as shown below.

➽ PROBLEM 1D: Problem Formulation in the Relative Frame

minimise $\int_0^{t_f} \sigma(t)\,dt$

subject to:

Kinematics:
$\dot{\boldsymbol{r}}^{\mathtt{A}}(t) = \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}}(t), \qquad \dot{\boldsymbol{q}}_{\mathtt{B/A}}(t) = \frac{1}{2}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}(t) \otimes \boldsymbol{q}_{\mathtt{B/A}}(t)$

Dynamics:
$\dot{z}(t) = -\alpha\sigma(t)$
$\dot{\boldsymbol{v}}^{\mathtt{A}}_{\mathtt{B/A}}(t) = \boldsymbol{\tau}^{\mathtt{A}}(t) + \boldsymbol{g}^{A} - 2\boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}(t)} \times \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}}(t) - \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}}(t) \times \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}}(t) \times \boldsymbol{r}^{\mathtt{A}}(t)$
$\dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) = \boldsymbol{J}^{-1}(\boldsymbol{T}^{\mathtt{B}}(t) - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}(t) \times \boldsymbol{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}(t) - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}(t) \times \boldsymbol{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}})(t) - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}}(t) \times \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}(t)$

Control Constraints:
$\|\boldsymbol{\tau}^{\mathtt{A}}(t)\| \leq \sigma(t)$
$\mu_1(t)\{1 - [z(t) - z_1(t)] + \frac{1}{2}[z_1(t) - z_2(t)]^2\} \leq \sigma(t) \leq \mu_2(t)\{1 - [z(t) - z_2(t)]\}$
$z_1(t) \leq z(t) \leq z_2(t)$

Boundary Conditions:
$m(0) = m_{wet}, \quad \boldsymbol{q}_{\mathtt{A/I}}(0) = \boldsymbol{q}_{\mathtt{A/I}_0}$
$\boldsymbol{r}^{\mathtt{A}}(0) = \boldsymbol{r}^{\mathtt{A}}_0, \quad \boldsymbol{r}^{\mathtt{A}}(t_f) = \boldsymbol{r}^{\mathtt{A}}_f, \quad \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}}(0) = \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}_0}, \quad \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}}(t_f) = \boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}_f}$
$\boldsymbol{q}_{\mathtt{B/A}}(0) = \boldsymbol{q}_{\mathtt{B/A}_0}, \quad \boldsymbol{q}_{\mathtt{B/A}}(t_f) = \boldsymbol{q}_{\mathtt{B/A}_f}, \quad \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}(0) = \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}_0}, \quad \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}(t_f) = \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}_f}$

Additional Constraints:
$\|A_j(\boldsymbol{x}) + b_j\| \leq c_j^T \boldsymbol{x} + d_j, \quad j = 1, \ldots, m$

6.2.3. Disadvantages of Lossless Convexification

The works of Blackmore and Açıkmeşe (2012), Açıkmeşe et al. (2013), Gerth (2014), Ridder (2016), etc., solve nonlinear optimal control problems using lossless convexification. All of them involve, solving a 3DOF problem with nonlinear system dynamics and nonconvex control constraints. But this method cannot guarantee a solution for problem structures that have nonlinear kinematics or nonconvex state constraints. To ensure that these nonlinearities and nonconvexities can also be handled, a new method has been researched and proved to provide convergence by making sure the convexified problem is always feasible and provides an optimal solution to the original problem. This method is *successive convexification*, which is the main focus of this thesis and has been explained in the next sections. Note that the problem formulation in Problem 1D for 6DOF has been presented for *lossless convexification* and can be researched by the interested reader to find if it can be solved by the method.

6.3. Successive Convexification

Mao et al. (2016) have presented a novel method to deal with such non-linear dynamics, *successive convexification*. The underlying logic is to linearise the nonconvexities in the dynamics and control constraints about the solution for that iteration, say k^{th} iteration. This leads to a convex sub-problem preferably in the SOCP

format, which can be solved to full optimality resulting in a new solution for the $(k+1)^{th}$ iterate. This process is continued successively till convergence. A continuous time convergence analysis done by Mao et al. (2016) guarantees convergence to the optimal solution of the original problem. The proof is too extensive to be explained in this thesis and can be referred to in Mao et al. (2016). The procedure for successive convexification will be explained step by step through the subsections below.

6.3.1. Linearisation

Problem 1D, shows the presence of non-linearities in the attitude kinematics, dynamics, translational dynamics, mass dynamics and thrust constraints. We deal with a non-linear polyhedron gravity field for asteroids, which adds non-linearity in the gravitational acceleration present in the translational dynamics and also a non-linearity due to the perturbing acceleration. All of these need to be convexified so to be suitable for the SOCP problem formulation. For this, we first redefine the state vectors discussed in Chapter 4 as below and also define the control vector.

$$\check{\boldsymbol{x}}_{\mathtt{A}}(t) = \begin{pmatrix} m(t) \\ \check{\boldsymbol{q}}_{\mathtt{B/A}}(t) \\ \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) \\ \check{\boldsymbol{F}}^{\mathtt{B}}(t) \\ \dot{\check{\boldsymbol{F}}}^{\mathtt{B}}(t) \end{pmatrix}, \quad \check{\boldsymbol{u}}_{\mathtt{A}}(t) = \left(\ddot{\check{\boldsymbol{F}}}^{\mathtt{B}}(t)\right) \tag{6.39}$$

The inclusion of the dual force and its first derivative in the state vector whilst the second derivative of the dual force in the control vector separates the state and control vector. This separation is beneficial, since it prevents high frequency jitters in control due to the non-linear dependency of the state on the control. The presence of these high frequency jitters have been found in Wang and Grant (2017) and their absence by separation has also been proved. Hence, we will go ahead with the separated vectors.

The convexification of the equality constraint is done by linearising them using first order Taylor approximation. But this does not ensure the optimal solution of the problem with linearised constraints to be the one for the original problem as well. To recover this optimality they are successively linearised till convergence. The continuous time non-linear dynamics is of the form,

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \quad a.e. \quad 0 \leq t \leq t_f \tag{6.40}$$

where, $\boldsymbol{x} : [0, t_f] \to \mathbb{R}^n$ represents the state trajectory, $\boldsymbol{u} : [0, t_f] \to \mathbb{R}^m$, the control inputs and $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$, the control-state mapping function. The control-state function will be assumed to be Fréchet differentiable [4] with respect to all arguments and the control input to be Lebesgue integrable, i.e., $\boldsymbol{u}$ is a measurable non negative decreasing function with a finite Lebesgue integral.[5] The first order Taylor expansion of the dynamics can be given as,

$$\begin{aligned} \dot{\boldsymbol{x}}(t) = f(\bar{\boldsymbol{x}}_{k-1}(t), \bar{\boldsymbol{u}}_{k-1}(t), t) + \frac{\partial}{\partial x} f(\boldsymbol{x}_{k-1}(t), \boldsymbol{u}_{k-1}(t), t)\Big(\boldsymbol{x}_{k-1}(t) - \bar{\boldsymbol{x}}_{k-1}(t)\Big) + \dots \\ + \frac{\partial}{\partial u} f(\boldsymbol{x}_{k-1}(t), \boldsymbol{u}_{k-1}(t), t)\Big(\boldsymbol{u}_{k-1}(t) - \bar{\boldsymbol{u}}_{k-1}(t)\Big) \end{aligned} \tag{6.41}$$

The above equation can be abbreviated with the following substitutions,

$$\boldsymbol{A}(t) = \frac{\partial}{\partial x} f(\boldsymbol{x}_{k-1}(t), \boldsymbol{u}_{k-1}(t), t)\Big|_{\bar{x}_{k-1}, \bar{u}_{k-1}} \tag{6.42}$$

$$\boldsymbol{B}(t) = \frac{\partial}{\partial u} f(\boldsymbol{x}_{k-1}(t), \boldsymbol{u}_{k-1}(t), t)\Big|_{\bar{x}_{k-1}, \bar{u}_{k-1}} \tag{6.43}$$

$$\boldsymbol{z}(t) = f(\bar{\boldsymbol{x}}_{k-1}(t), \bar{\boldsymbol{u}}_{k-1}(t), t) - A(t)\bar{\boldsymbol{x}}_{k-1}(t) - B(t)\bar{\boldsymbol{u}}_{k-1}(t) \tag{6.44}$$

The $\boldsymbol{A}(t)$ matrix is called the *system matrix* and the $\boldsymbol{B}(t)$ the *control matrix*. The linearised dynamics would, therefore, look like,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{k-1}(t)\boldsymbol{x}_{k-1}(t) + \boldsymbol{B}_{k-1}(t)\boldsymbol{u}_{k-1}(t) + \boldsymbol{z}_{k-1}(t) \tag{6.45}$$

The problem formulation as shown in Problem 1E is as per lossless convexification and includes slack variables to convexify the nonconvex controls. However, the nonconvexity in the dynamics still exists. We can

[4] A function f is Fréchet differentiable at a if $\lim_{x \to a} = \frac{f(x)-f(a)}{x-a}$ exists.
[5] Lebesgue integral is by definition $\int_0^\infty u(t)dt$.

linearise all the non-convex constraints and dynamics in the successive convexification algorithm, or we can include the slack variables for the convexified constraints. From the state vector and control vector in Eq. (6.39) we can state the dynamic state as shown in Eq. (6.46).

The approach is to directly linearise the non-linearities in the dynamic state at the initial and final time about $\bar{\boldsymbol{x}}(t)$. The state differential, from the relative state kinematics and dynamics, as discussed in Chapter 4 is reiterated here. The partial differentials with respect to the state vector, to linearise the non-linear dynamics, using Taylor series expansion of the first order have been derived here. These derivations, for a dual quaternion state in the relative frame, are a novelty of this thesis. As per the information available in the duration of this thesis the Jacobian matrix for such a state has been derived for the first time.

$$\dot{\check{\boldsymbol{x}}}_{\mathtt{A}}(t) = \begin{pmatrix} -\alpha\|\boldsymbol{F}^{\mathtt{B}}(t)\| \\ \frac{1}{2}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t)\check{\otimes}\check{\boldsymbol{q}}_{\mathtt{B/A}}(t) \\ \check{\boldsymbol{J}}^{-1}(\check{\boldsymbol{F}}^{\mathtt{B}}(t) + \check{\boldsymbol{F}}^{\mathtt{B}}_{D}(t) - (\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t))\check{\times}\check{\boldsymbol{J}}(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t)) - \check{\boldsymbol{J}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t)\check{\times}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) - \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t)\check{\times}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t)\check{\times}\boldsymbol{R}(t)) \\ \dot{\check{\boldsymbol{F}}}^{\mathtt{B}}(t) \\ \check{\boldsymbol{u}}_{\mathtt{B}}(t) \end{pmatrix} \tag{6.46}$$

$$f_{\boldsymbol{F}^{\mathtt{B}},t} = \left(\|\bar{\boldsymbol{F}}^{\mathtt{B}}(t)\| + \frac{\bar{\boldsymbol{F}}^{\mathtt{B}}(t)}{\|\bar{\boldsymbol{F}}^{\mathtt{B}}(t)\|}\left(\boldsymbol{F}^{\mathtt{B}}(t) - \bar{\boldsymbol{F}}^{\mathtt{B}}(t)\right)\right) \tag{6.47}$$

$$f_{\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}},t} = \dot{\bar{\check{\boldsymbol{q}}}}_{\mathtt{B/A}}(t) + \left.\frac{\partial\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}}}{\partial\boldsymbol{x}}\right|_{\bar{\check{\boldsymbol{x}}}_{\mathtt{A}}(t)}(\boldsymbol{x}_{\mathtt{A}}(t) - \bar{\check{\boldsymbol{x}}}_{\mathtt{A}}(t)) \tag{6.48}$$

$$f_{\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}},t} = \dot{\bar{\check{\boldsymbol{\omega}}}}^{\mathtt{B}}_{\mathtt{B/A}}(t) + \left.\frac{\partial\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}}{\partial\boldsymbol{x}}\right|_{\bar{\check{\boldsymbol{x}}}_{\mathtt{A}}(t)}(\boldsymbol{x}_{\mathtt{A}}(t) - \bar{\check{\boldsymbol{x}}}_{\mathtt{A}}(t)) \tag{6.49}$$

For the disturbance forces and torques in the attitude and translational dynamics, we will consider only the gravitational acceleration and gravity gradient torque, since in close proximity to the asteroid the SRP and 3rd body forces and torques can be assumed to be non-existent. They are negligible as compared to the effects due to gravity. To obtain the linearisation matrices in continuous time, we first derive the partial differentials with respect to state and controls step by step for the kinematics and dynamics.

Mass Dynamics:

$$\dot{m}(t) = -\alpha\left(\|\bar{\boldsymbol{F}}^{\mathtt{B}}(t)\| + \frac{\bar{\boldsymbol{F}}^{\mathtt{B}}(t)}{\|\bar{\boldsymbol{F}}^{\mathtt{B}}(t)\|}\left(\boldsymbol{F}^{\mathtt{B}}(t) - \bar{\boldsymbol{F}}^{\mathtt{B}}(t)\right)\right) \tag{6.50}$$

The partial differential becomes,

$$\boldsymbol{\xi}_{\check{F}} = \begin{bmatrix} -\alpha\frac{(\bar{\boldsymbol{F}}^{\mathtt{B}})'(t)}{\|\bar{\boldsymbol{F}}^{\mathtt{B}}(t)\|} & \boldsymbol{0}_{1\times5} \end{bmatrix} \tag{6.51}$$

Kinematics:

$$\begin{aligned} \dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}}(t) &= \frac{1}{2}\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t)\check{\otimes}\right]\check{\boldsymbol{q}}_{\mathtt{B/A}}(t) \\ &= \frac{1}{2}\begin{bmatrix} [\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}\otimes] & 0_{4\times4} \\ [\boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}}\otimes] & [\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}}\otimes] \end{bmatrix}\begin{pmatrix} \boldsymbol{q}_{\mathtt{B/A}} \\ \boldsymbol{q}_{d_{\mathtt{B/A}}} \end{pmatrix} \\ &= \frac{1}{2}\begin{bmatrix} [\boldsymbol{q}_{\mathtt{B/A}}\odot] & 0_{4\times4} \\ [\boldsymbol{q}_{d_{\mathtt{B/A}}}\odot] & [\boldsymbol{q}_{\mathtt{B/A}}\odot] \end{bmatrix}\begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} \end{pmatrix} \end{aligned} \tag{6.52}$$

We find the partial differentials below,

$$\check{\boldsymbol{\omega}}'_{\mathtt{B/A}} = \left.\frac{\partial\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}}{\partial\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}}\right|_{\check{\check{\boldsymbol{x}}}} = \left[\begin{array}{cc|cc} \boldsymbol{I}_3 & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \\ \hline \boldsymbol{0}_{3\times3} & 0 & \boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \tag{6.53}$$

$$\boldsymbol{\Psi}_{\check{q}} = \left.\frac{\partial\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}}}{\partial\check{\boldsymbol{q}}_{\mathtt{B/A}}}\right|_{\check{\check{\boldsymbol{x}}}} = 0.5\begin{bmatrix} [\bar{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}\otimes] & \boldsymbol{0}_{4\times4} \\ [\bar{\boldsymbol{v}}^{\mathtt{B}}_{\mathtt{B/A}}\otimes] & [\bar{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}\otimes] \end{bmatrix}\boldsymbol{I}_8 \tag{6.54}$$

$$\boldsymbol{\Psi}_{\check{\omega}} = \left.\frac{\partial\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}}}{\partial\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}}\right|_{\check{\check{\boldsymbol{x}}}} = 0.5\begin{bmatrix} [\bar{\boldsymbol{q}}_{\mathtt{B/A}}\odot] & \boldsymbol{0}_{4\times4} \\ [\bar{\boldsymbol{q}}_{d_{\mathtt{B/A}}}\odot] & [\bar{\boldsymbol{q}}_{\mathtt{B/A}}\odot] \end{bmatrix}\check{\boldsymbol{\omega}}'_{\mathtt{B/A}} \tag{6.55}$$

Dynamics:

$$\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}(t) = \check{\boldsymbol{J}}^{-1}\big(\check{\boldsymbol{F}}^{\mathtt{B}}(t) + \check{\boldsymbol{F}}^{\mathtt{B}}_{D}(t) - (\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t)) \check{\times} \check{\boldsymbol{J}}(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t)) - \check{\boldsymbol{J}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t) \check{\times} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) \\ - \check{\boldsymbol{J}}(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t) \check{\times} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}(t) \check{\times} \boldsymbol{R}(t))\big) \tag{6.56}$$

$$= \check{\boldsymbol{J}}^{-1}\Bigg(\begin{pmatrix} \boldsymbol{F}^{\mathtt{B}} \\ 0 \\ \boldsymbol{r}_{ex} \times \boldsymbol{F}^{\mathtt{B}} \\ 0 \end{pmatrix} + \begin{pmatrix} m\boldsymbol{g}^{\mathtt{B}} \\ 0 \\ \boldsymbol{T}^{\mathtt{B}}_{GG} \\ 0 \end{pmatrix} - \left(\begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}} \end{pmatrix}\right) \check{\times} \check{\boldsymbol{J}}\left(\begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}} \end{pmatrix}\right) - \check{\boldsymbol{J}}\begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}} \end{pmatrix} \check{\times} \begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/A}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}} \end{pmatrix}$$

$$- \check{\boldsymbol{J}}\begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}} \end{pmatrix} \check{\times} \begin{pmatrix} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{A/I}} \\ \boldsymbol{v}^{\mathtt{B}}_{\mathtt{A/I}} \end{pmatrix} \check{\times} \boldsymbol{R}\Bigg) \tag{6.57}$$

$$= \check{\boldsymbol{J}}^{-1}\boldsymbol{W} \tag{6.58}$$

The partial differentials with respect to mass,

$$\check{\boldsymbol{J}}' = \frac{\partial \check{\boldsymbol{J}}}{\partial m}\bigg|_{\check{\check{\boldsymbol{x}}}} = \left[\begin{array}{cc|cc} \boldsymbol{0}_{3\times3} & 0 & \boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & 0 \\ \hline \boldsymbol{0}_{3\times3} & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \end{array}\right], \qquad \check{\boldsymbol{J}}'' = \frac{\partial \check{\boldsymbol{J}}^{-1}}{\partial m}\bigg|_{\check{\check{\boldsymbol{x}}}} = \left[\begin{array}{cc|cc} \boldsymbol{0}_{3\times3} & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \\ \hline \frac{-1}{m^2}\boldsymbol{I}_3 & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \tag{6.59}$$

$$\boldsymbol{\Theta}_m = \frac{\partial \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}}{\partial m}\bigg|_{\check{\check{\boldsymbol{x}}}} = \check{\boldsymbol{J}}^{-1}\left(\begin{pmatrix} \boldsymbol{g}^{\mathtt{B}} \\ 0 \\ \sum_{i=1}^{n} \boldsymbol{\rho}_i \times \boldsymbol{g}^{\mathtt{B}}_i \\ 0 \end{pmatrix} - (\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}) \check{\times} \check{\boldsymbol{J}}'(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}) - \check{\boldsymbol{J}}'(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}} \check{\times} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}) - \check{\boldsymbol{J}}'(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}} \check{\times} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}} \check{\times} \bar{\boldsymbol{R}})\right)$$

$$+ \check{\boldsymbol{J}}''\bar{\boldsymbol{W}} \tag{6.60}$$

The partial differentials with respect to the dual quaternion,

$$\check{\boldsymbol{\omega}}'_{\mathtt{A/I}} = \frac{\partial \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}}{\partial \check{\boldsymbol{q}}_{\mathtt{B/A}}}\bigg|_{\check{\check{\boldsymbol{x}}}} = 2\left[\begin{array}{cc|cc} \left[\left(\boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{B/A}} \otimes \boldsymbol{q}^{*}_{\mathtt{B/A}}\right)\odot\right] & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \\ \hline \left[\left(\boldsymbol{v}^{\mathtt{A}}_{\mathtt{B/A}} \otimes \boldsymbol{q}^{*}_{\mathtt{B/A}}\right)\odot\right] & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \tag{6.61}$$

$$\boldsymbol{R}' = \frac{\partial \boldsymbol{R}}{\partial \check{\boldsymbol{q}}_{\mathtt{B/A}}}\bigg|_{\check{\check{\boldsymbol{x}}}} = 2\begin{bmatrix} \boldsymbol{0}_{4\times4} & \boldsymbol{0}_{4\times4} \\ [-1\ -1\ -1\ 1]^{\mathrm{T}}\left[\boldsymbol{q}_{\mathtt{B/A}_d}\otimes\right] & \left[\boldsymbol{q}^{*}_{\mathtt{B/A}}\otimes\right] \end{bmatrix} \tag{6.62}$$

$$\boldsymbol{\Theta}_{\check{q}} = \frac{\partial \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}}{\partial \check{\boldsymbol{q}}_{\mathtt{B/A}}}\bigg|_{\check{\check{\boldsymbol{x}}}} = \check{\check{\boldsymbol{J}}}^{-1}\big(\check{\boldsymbol{F}}'_{D} - \left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}\check{\times}\right]\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}'_{\mathtt{A/I}} + \left[\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{A/I}} - \left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}'_{\mathtt{A/I}} + \left[\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{A/I}}$$

$$+ \check{\check{\boldsymbol{J}}}\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{A/I}} - \check{\check{\boldsymbol{J}}}\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\boldsymbol{R}' + \check{\check{\boldsymbol{J}}}\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right](\left[\bar{\boldsymbol{R}}\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{A/I}})$$

$$- \check{\check{\boldsymbol{J}}}\left[\left(\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\bar{\boldsymbol{R}}\right)\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{A/I}}\big) \tag{6.63}$$

The partial differentials with respect to dual velocity,

$$\boldsymbol{\Theta}_{\check{\omega}} = \frac{\partial \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}}{\partial \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}}\bigg|_{\check{\check{\boldsymbol{x}}}} = \check{\check{\boldsymbol{J}}}^{-1}\left(-\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}\check{\times}\right]\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}'_{\mathtt{B/A}} + \left[\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{B/A}} + \left[\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{B/A}} - \left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\check{\check{\boldsymbol{J}}}\check{\boldsymbol{\omega}}'_{\mathtt{B/A}}\check{\check{\boldsymbol{J}}}\left[\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}\check{\times}\right]\check{\boldsymbol{\omega}}'_{\mathtt{B/A}}\right) \tag{6.64}$$

The partial differentials with respect to dual force,

$$\boldsymbol{\Theta}_{\check{F}} = \frac{\partial \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}}{\partial \check{\boldsymbol{F}}^{\mathtt{B}}}\bigg|_{\check{\check{\boldsymbol{x}}}} = \check{\check{\boldsymbol{J}}}^{-1}\frac{\partial \check{\boldsymbol{F}}^{\mathtt{B}}}{\partial \check{\boldsymbol{F}}^{\mathtt{B}}}\bigg|_{\check{\check{\boldsymbol{x}}}} = \check{\check{\boldsymbol{J}}}^{-1}\left[\begin{array}{cc|cc} \boldsymbol{I}_3 & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \\ \hline \left[\boldsymbol{r}_e\times\right] & 0 & \boldsymbol{0}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \tag{6.65}$$

Polyhedral Gravity Field:
The polyhedral gravity field is discussed in Section 4.11.2. We will reiterate the analytical equation for gravity and the gravity gradient torque here,

$$\begin{aligned} \boldsymbol{g}^{\mathrm{B}} &= \nabla U^{\mathrm{B}} = \boldsymbol{q}_{\mathrm{B/A}} \otimes \left(-G\,\varrho \sum_{e \in edges} \boldsymbol{E}_e\, \boldsymbol{r}_e\, L_e + G\,\varrho \sum_{f \in faces} \boldsymbol{F}_f\, \boldsymbol{r}_f\, \omega_f \right) \otimes \boldsymbol{q}^{*}_{\mathrm{B/A}} \\ \boldsymbol{T}^{\mathrm{B}}_{GG} &= \sum_{i=1}^{n} \boldsymbol{\rho}_i \times m_i\, \boldsymbol{g}^{\mathrm{B}}_i \end{aligned} \tag{6.66}$$

Due to the complexity and highly non-linear dependence of the polyhedral gravity to the position vector of the SC, analytical partial differentiation is out of scope of this thesis. We use numerical differentiation for the purpose of this thesis to obtain the partial differentials of the gravitational dual force vector with respect to the relative dual quaternion, $\check{\boldsymbol{q}}_{\mathrm{B/A}}$. From the overview of these analytic functions, it can be seen, that they involve a combination of algebraic, trigonometric, logarithmic and second norm functions, which are all holomorphic in $\mathbb{C}$ and therefore, we go ahead with complex-step derivative approximation for the first order partial derivatives. A detailed discussion of the method and the comparison of approximations with finite difference is also discussed.

- **Complex-Step Derivative Approximation**
  The classical methods of finite difference approximations fall short of accurately representing the partial differentiation of complex non-linear analytical functions, since they are ill-conditioned. Complex step differentiation method is better suited to handle such non-linear partial derivatives. Lyness and Moler (1967) used complex variables to develop formulae for numerical differentiation and numerical analysis for the same was done by Squire and Trapp (1998), which can be referred to, for a more detailed description. This method avoids the subtractive cancellation errors involved in the classical numerical differentiation methods, since it does not deal with a subtraction between perturbed function values.

  A *holomorphic* function, f in the open set U(where $U \subset \mathbb{C}$), is a complex-valued function of one or more complex variables, such that it is complex differentiable at any point in its domain[6]. In a broader sense, it is also referred to as an *analytic* function, which applies continuous differentiability in the real plane as well, such that the function can be written as a convergent Taylor series at every point in its domain. This is a requirement for complex numerical differentiation. The analytic function is decomposed to take the following form (Martins et al. (2003)),

$$f(x+iy) = u(x,y) + i\,v(x,y) \tag{6.67}$$

  Cauchy-Riemann equations provide the relationship between the real and complex differentiability, which in turn helps establish if the function is holomorphic. These equations are given as (Martins et al. (2003)),

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \tag{6.68}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \tag{6.69}$$

  The RHS of the first equation Cauchy-Riemann is used to derive the equation for the complex-step derivative approximation, which is given as (Martins et al. (2003)),

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{\mathrm{Im}[f(x+ih)]}{h} \tag{6.70}$$

  which for a small discrete h can be approximated as (Martins et al. (2003)),

$$\frac{\partial f}{\partial x} \approx \frac{\mathrm{Im}[f(x+ih)]}{h} \tag{6.71}$$

  The analytic function for polyhedral gravity and the gravity gradient torque need to be proved as holomorphic functions to be applied for complex step derivative approximations. For this thesis, we leave the analysis of the polyhedral functions by the Cauchy-Riemann equations for future work, since Werner and Scheeres (1997) state that these functions are analytic and are second differentiable. The verification for the correct implementation of the complex-step derivative approximation has been provided in Section 7.6.1.

[6] https://en.wikipedia.org/wiki/Holomorphic_function, date accessed:02-10-18

6.3.2. Continuous-Time State and Control Matrices

The continuous time matrices, $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$ and $\boldsymbol{z}(t)$ are given below. As can be seen, the system of equations are *linear time variant* (LTV) and this needs to be treated by discretisation to achieve the discrete time LTV matrices, which will be discussed in the upcoming sections. This approach directly linearises the kinematics and dynamics at the initial and final time about $\bar{\boldsymbol{x}}(t)$.

$$\boldsymbol{A}(t) = \begin{bmatrix} 0 & \mathbf{0}_{1\times 8} & \mathbf{0}_{1\times 8} & \boldsymbol{\xi}_{\check{F}} & \mathbf{0}_{1\times 8} \\ \mathbf{0}_{8\times 1} & \boldsymbol{\Psi}_{\check{q}} & \boldsymbol{\Psi}_{\check{\omega}} & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 8} \\ \boldsymbol{\Theta}_m & \boldsymbol{\Theta}_{\check{q}} & \boldsymbol{\Theta}_{\check{\omega}} & \boldsymbol{\Theta}_{\check{F}} & \mathbf{0}_{8\times 8} \\ \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 8} & \boldsymbol{I}_{8\times 8} \\ \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 8} & \mathbf{0}_{8\times 8} \end{bmatrix}, \qquad \boldsymbol{B}(t) = \begin{bmatrix} \mathbf{0}_{25\times 8} \\ \boldsymbol{I}_8 \end{bmatrix}, \tag{6.72}$$

$$\boldsymbol{z}(t) = \begin{pmatrix} \dot{\bar{m}}(t) \\ \dot{\bar{\boldsymbol{q}}}_{\mathtt{B/A}}(t) \\ \dot{\bar{\boldsymbol{q}}}_{d_{\mathtt{B/A}}}(t) \\ \dot{\bar{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}(t) \\ 0 \\ \dot{\bar{\boldsymbol{v}}}^{\mathtt{B}}_{\mathtt{B/A}}(t) \\ 0 \\ \dot{\bar{\boldsymbol{F}}}^{\mathtt{B}}(t) \\ 0 \\ \dot{\bar{\boldsymbol{T}}}^{\mathtt{B}}(t) \\ 0 \\ \ddot{\bar{\boldsymbol{F}}}^{\mathtt{B}}(t) \\ 0 \\ \ddot{\bar{\boldsymbol{T}}}^{\mathtt{B}}(t) \\ 0 \end{pmatrix} - \boldsymbol{A}(t) \begin{pmatrix} \bar{m}(t) \\ \bar{\boldsymbol{q}}_{\mathtt{B/A}}(t) \\ \bar{\boldsymbol{q}}_{d_{\mathtt{B/A}}}(t) \\ \bar{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}}(t) \\ 0 \\ \bar{\boldsymbol{v}}^{\mathtt{B}}_{\mathtt{B/A}}(t) \\ 0 \\ \bar{\boldsymbol{F}}^{\mathtt{B}}(t) \\ 0 \\ \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \\ 0 \\ \ddot{\bar{\boldsymbol{F}}}^{\mathtt{B}}(t) \\ 0 \\ \ddot{\bar{\boldsymbol{T}}}^{\mathtt{B}}(t) \\ 0 \end{pmatrix} - \boldsymbol{B}(t) \begin{pmatrix} \ddot{\bar{\boldsymbol{F}}}^{\mathtt{B}}(t) \\ 0 \\ \ddot{\bar{\boldsymbol{T}}}^{\mathtt{B}}(t) \\ 0 \end{pmatrix} \tag{6.73}$$

Now we reformulate the problem in the relative frame as discussed in lossless convexification (Problem 1D) to one using linearised DQ dynamics and kinematics and SOCP constraints.

➽ PROBLEM 1E: Problem Formulation using Dual Quaternions

minimise $-m_{t_f}$

subject to:

Kinematics and Dynamics:
$\dot{m}(t) = -\alpha f_{\boldsymbol{F}^{\mathtt{B}},k}$

$\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}} = f_{\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}},k}, \qquad \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}} = f_{\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}},k}$

Control Constraints:
$m(t_f) \leq m_k$
$F_{min} \leq f_{\boldsymbol{F}^{\mathtt{B}},k} \leq F_{max}$

Boundary Conditions:
$m_0 = m_{wet}, \quad \boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{A/I}} = \text{constant}$
$\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}}(0) = \dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}_0}, \quad \dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}}(t_f) = \dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}_0}, \quad \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}(0) = \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}_0}, \quad \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}}(t_f) = \dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}_f}$

Additional Constraints:
$\|A_j(\boldsymbol{x}) + b_j\|_2 \leq c_j^T \boldsymbol{x} + d_j, \quad j = 1, \ldots, m$

6.3.3. Discretisation

The differential of the state represents continuous-time dynamics, and this needs to be converted to a finite dimensional parameter optimisation problem. The trajectory from initial to the final state can, therefore, be discretised to K points in the total time. The final time, t_f can, therefore, be divided into K nodes and the

time at each node is indexed by $k \in \mathbb{N}$ and can be given as,

$$t_k \triangleq \Big(\frac{k}{K-1}\Big) t_f, \quad k \in [0, K) \tag{6.74}$$

The sampling period, Δt is equal to $\frac{t_f}{K-1}$. The initial time can be $t_0 = 0$ s without any loss of generality, since the problem is not explicitly time-dependent.

Control Parametrisation

A parametrisation of the control vector is needed to map the control variable infinite time for the solver to give optimal controls to be used in continuous time by the actual system. This can be done via a zero-order (ZOH) or first-order (FOH) hold on the control. Other methods include the use of Chebyshev polynomials, but to keep it simple we approach the problem with ZOH. The continuous time control can be represented as a linear combination of basis functions, ϑ_k,

$$\check{\boldsymbol{u}}(t) = \sum_{k=1}^{K} \check{\boldsymbol{u}}_k \, \vartheta_k(t) \tag{6.75}$$

where, $\check{\boldsymbol{u}}_k \triangleq \check{\boldsymbol{u}}(t_k) \;\forall\; k \in [0, K)$ are the optimal controls given by the solver. For ZOH, the basis functions are defined as,

$$\vartheta_k(t) = \begin{cases} 1 & \text{when } t = [t_{k-1}, t_k] \\ 0 & \text{otherwise} \end{cases} \tag{6.76}$$

The above definition means, that the basis functions are active ($\vartheta = 1$) from t_{k-1} to t_k, but not including t_k. This keeps the optimal control, $\check{\boldsymbol{u}}_k$ during the interval and deactivates it at the next time step. Since, the descent problem is a minimum fuel problem, a bang-bang control profile is expected and hence ZOH is suitable for the algorithm. The black discontinuous lines in Figure 6.8 represents how the control output looks like with respect to time, with ZOH. However, as can be seen, ZOH is a stepping function and errors are introduced, since the discontinuities are not physically possible to be implemented by the actual system.

The FOH on the controls is a piecewise linear approximation of the actual required control and below we mention the predictive FOH representative basis functions,

$$\vartheta_k(t) = \begin{cases} \dfrac{t_{k+1} - t}{t_{k+1} - t_k} & \text{when } t = [t_{k-1}, t_k] \\ \dfrac{t - t_k}{t_{k+1} - t_k} & \text{when } t = [t_k, t_{k+1}] \\ 0 & \text{otherwise} \end{cases} \tag{6.77}$$

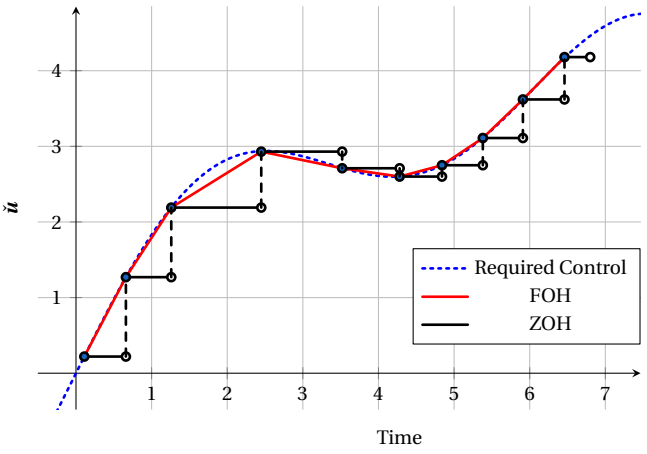


Figure 6.8: Actual required control output is depicted by the blue dotted line and (a) The discontinuous black lines represent the ZOH function. (b) The continuous red lines represent the FOH function

The output obtained from a FOH on control is shown in Figure 6.8, with the red continuous line between the points. The control vector for FOH looks like,

$$\check{\boldsymbol{u}}(t) = \vartheta_k^-(t)\check{\boldsymbol{u}}_k + \vartheta_k^+(t)\boldsymbol{u}_{k+1} \tag{6.78}$$

In the 6DOF Mars descent problem, Szmuk et al. (2017) have applied FOH for the control to prevent the errors due to the discontinuities in ZOH. Ridder (2016) has advised from his research to implement FOH for convex guidance to ensure a smooth attitude command. He uses a trajectory tracker, that picks up deviations from the optimal trajectory to reduce the severity of a direct ZOH controller. Due to the complexity of the problem at hand, for this thesis, we will approach the problem with ZOH. It is shown, in Section 7.5.2, in the verification for the SCvx algorithm using the 6DOF Mars test case, that the control results are very close to what they have achieved with ZOH. However, as a recommendation, further research work should be towards either implementing FOH or a trajectory tracker for a smooth attitude command.

Continuous and Discrete Time State Space Solution

As we have already established in the previous section, that the system we have at hand is an LTV, we need to find the solution likewise. The general solution for a linear time-varying system in continuous time with ZOH on the control can be found in Antsaklis and Michel (2006) and is given as,

$$\underbrace{\boldsymbol{\phi}(t, t_0, x_0) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}_0}_{\substack{\text{homogeneous} \\ \text{solution}}} + \underbrace{\int_{t_0}^{t} \boldsymbol{\Phi}(t, s)\boldsymbol{B}(s)\boldsymbol{u}(s)\mathrm{d}s + \int_{t_0}^{t} \boldsymbol{\Phi}(t, s)\boldsymbol{z}(s)\mathrm{d}s}_{\substack{\text{particular} \\ \text{solution}}} \tag{6.79}$$

where, $\boldsymbol{\Phi}(t, t_0)$ is the state transition matrix for zero-input evolution from the previous to next state and $\boldsymbol{\Phi}(t, t_0) \in \mathbb{R}^{n\times n}$. This equation needs to be solved as is, if we do not discretise the problem. Discretisation enables a simpler solution for the already complex system of dynamics and kinematics. The discretised system and control matrix will be denoted as $\tilde{\boldsymbol{A}}(t)$, $\tilde{\boldsymbol{B}}(t)$ and $\tilde{\boldsymbol{z}}(t)$. The state transition matrix for a discretised non-homogeneous LTV system is given in Antsaklis and Michel (2006) as,

$$\tilde{\boldsymbol{\Phi}}(k, k_0) = \prod_{i=k_0}^{k-1} \tilde{\boldsymbol{A}}_i,\ i \geq k_0,\ k \neq k_0 \tag{6.80}$$

where, $\tilde{\boldsymbol{\Phi}}(k_0, k_0) = \boldsymbol{I}_n$. The state space solution for the discrete time system, therefore, can be given by,

$$\boldsymbol{x}_k = \tilde{\boldsymbol{\Phi}}(k, k_0)\boldsymbol{x}_0 + \sum_{i=k_0}^{k-1} \tilde{\boldsymbol{\Phi}}(k, i+1)\tilde{\boldsymbol{B}}_i\boldsymbol{u}_i + \sum_{i=k_0}^{k-1} \tilde{\boldsymbol{\Phi}}(k, i+1)\tilde{\boldsymbol{z}}_i \tag{6.81}$$

As an example, for $t = t_2$, $k = 2$, the state $\boldsymbol{x}_2$ can be given as,

$$\begin{aligned}
\boldsymbol{x}_2 &= \tilde{\boldsymbol{\Phi}}(2,0)\boldsymbol{x}_0 + \sum_{i=0}^{1} \tilde{\boldsymbol{\Phi}}(2, i+1)\tilde{\boldsymbol{B}}_i\boldsymbol{u}_i + \sum_{i=0}^{1} \tilde{\boldsymbol{\Phi}}(2, i+1)\tilde{\boldsymbol{z}}_i \\
&= \tilde{\boldsymbol{\Phi}}(2,0)\boldsymbol{x}_0 + \tilde{\boldsymbol{\Phi}}(2,1)\tilde{\boldsymbol{B}}_0\boldsymbol{u}_0 + \tilde{\boldsymbol{\Phi}}(2,2)\tilde{\boldsymbol{B}}_1\boldsymbol{u}_1 + \tilde{\boldsymbol{\Phi}}(2,1)\tilde{\boldsymbol{z}}_0 + \tilde{\boldsymbol{\Phi}}(2,2)\tilde{\boldsymbol{z}}_1 \\
&= \tilde{\boldsymbol{A}}_1\tilde{\boldsymbol{A}}_0\boldsymbol{x}_0 + \tilde{\boldsymbol{A}}_1\tilde{\boldsymbol{B}}_0\boldsymbol{u}_0 + \tilde{\boldsymbol{B}}_1\boldsymbol{u}_1 + \tilde{\boldsymbol{A}}_1\tilde{\boldsymbol{z}}_0 + \tilde{\boldsymbol{z}}_1
\end{aligned}$$

For *linear time invariant* (LTI) systems, the discrete state matrix and control matrix are given by Antsaklis and Michel (2006),

$$\begin{aligned}
\boldsymbol{\Upsilon}(\Delta t) &= \sum_{j=0}^{\infty} \left(\frac{\Delta t^j}{(j+1)!}\right)\boldsymbol{A}^j = \boldsymbol{I}_n + \frac{\Delta t}{2!}\boldsymbol{A} + \frac{\Delta t^2}{3!}\boldsymbol{A}^2 + \frac{\Delta t^3}{2!}\boldsymbol{A}^3 + \mathcal{O}(\boldsymbol{A}^4) \\
\tilde{\boldsymbol{A}} &= e^{\boldsymbol{A}\Delta t} = \boldsymbol{I}_n + \Delta t\ \boldsymbol{A}\ \boldsymbol{\Upsilon}(\Delta t) \\
\tilde{\boldsymbol{B}} &= \int_0^{\Delta t} e^{\boldsymbol{A}\Delta t_s}\boldsymbol{B}\ \mathrm{d}s = \left(\sum_{j=0}^{\infty}\left(\frac{\Delta t^{j+1}}{(j+1)!}\right)\boldsymbol{A}^j\right)\boldsymbol{B} = \Delta t\ \boldsymbol{\Upsilon}(\Delta t)\boldsymbol{B} \\
\tilde{\boldsymbol{z}} &= \int_0^{\Delta t} e^{\boldsymbol{A}\Delta t_s}\boldsymbol{z}\ \mathrm{d}s = \left(\sum_{j=0}^{\infty}\left(\frac{\Delta t^{j+1}}{(j+1)!}\right)\boldsymbol{A}^j\right)\boldsymbol{z} = \Delta t\ \boldsymbol{\Upsilon}(\Delta t)\boldsymbol{z}
\end{aligned} \tag{6.82}$$

Extending the same for the LTV system where the discrete system and control matrices have to be calculated at each time step,

$$\begin{aligned}\boldsymbol{\Upsilon}(\Delta t,k)=\sum_{j=0}^{\infty}\left(\frac{\Delta t^j}{(j+1)!}\right)\boldsymbol{A}_k^j=\boldsymbol{I}_n+\frac{\Delta t}{2!}\boldsymbol{A}_k+\frac{\Delta t^2}{3!}\boldsymbol{A}_k^2+\frac{\Delta t^3}{2!}\boldsymbol{A}_k^3+\mathcal{O}(\boldsymbol{A}_k^4)\\ \tilde{\boldsymbol{A}}_k=\boldsymbol{I}_n+\Delta t\ \boldsymbol{A}_k\boldsymbol{\Upsilon}(\Delta t)\\ \tilde{\boldsymbol{B}}_k=\Delta t\ \boldsymbol{\Upsilon}(\Delta t,k)\boldsymbol{B}_k\\ \tilde{\boldsymbol{z}}_k=\Delta t\ \boldsymbol{\Upsilon}(\Delta t,k)\boldsymbol{z}_k\end{aligned}\tag{6.83}$$

The linear discrete time variant dynamics for the first order can be given as,

$$\boldsymbol{x}_{k+1}=\tilde{\boldsymbol{A}}_k\boldsymbol{x}_k+\tilde{\boldsymbol{B}}_k\boldsymbol{u}_k+\tilde{\boldsymbol{z}}_k\tag{6.84}$$

while that for higher order is given as,

$$\boldsymbol{x}_{k+1}=\tilde{\boldsymbol{A}}_k\boldsymbol{x}_k+\tilde{\boldsymbol{A}}_k\tilde{\boldsymbol{A}}_{k-1}\dots\tilde{\boldsymbol{A}}_1\begin{pmatrix}\tilde{\boldsymbol{B}}_0\boldsymbol{u}_0\\ \tilde{\boldsymbol{z}}_0\end{pmatrix}+\tilde{\boldsymbol{A}}_k\tilde{\boldsymbol{A}}_{k-1}\dots\tilde{\boldsymbol{A}}_2\begin{pmatrix}\tilde{\boldsymbol{B}}_1\boldsymbol{u}_1\\ \tilde{\boldsymbol{z}}_1\end{pmatrix}+\dots+\tilde{\boldsymbol{A}}_k\tilde{\boldsymbol{A}}_{k-1}\begin{pmatrix}\tilde{\boldsymbol{B}}_k\boldsymbol{u}_k\\ \tilde{\boldsymbol{z}}_k\end{pmatrix}\tag{6.85}$$

6.3.4. Cost Function

The original OCP is discussed earlier in Eq. (6.1). The cost function of the original problem looks like,

$$\text{P1: minimise } f(\boldsymbol{x},\boldsymbol{u})\tag{6.86}$$

The cost function has to be penalised for the constrained problem, which includes the dynamic equality constraints for each time step and the linear inequality state and control controls, also for each time step. The cost function, therefore, becomes,

$$\text{P2: minimise } \boldsymbol{N}_c(\boldsymbol{x},\boldsymbol{u}):=f(\boldsymbol{x},\boldsymbol{u})+w_h\|h_l(\boldsymbol{x},\boldsymbol{u})\|_1+w_g\sum_{j=1}^{m}\max\{0,g_j(\boldsymbol{x},\boldsymbol{u})\}\tag{6.87}$$

Introducing the virtual controls (discussed further in the next section), $\boldsymbol{\nu}$ and slacks variables for the constraints, $\boldsymbol{\zeta}$ to form an equivalent relaxed penalised problem, such that the solution is the same as P2.

$$\text{P3: minimise } f(\boldsymbol{x},\boldsymbol{u})+w_\nu\sum_{l=1}^{n}\|\boldsymbol{\nu}_l\|_2+w_g\sum_{j=1}^{m}\boldsymbol{\zeta}_j\tag{6.88}$$

Making the penalty weights sufficiently large, can ensure, that the problems P2 and P3 represent the same problem as P1, so as to obtain the solution to the original problem. The next step is to linearise the cost function to approximate the non-linear cost. A Taylor series expansion of the first order of P2 is used to do this (Wang and Cui (2018)).

$$\text{LP2: minimise } \boldsymbol{L}_c(\boldsymbol{x},\boldsymbol{u}):=f(\boldsymbol{x},\boldsymbol{u})+\nabla f(\boldsymbol{x},\boldsymbol{u}).\delta\boldsymbol{x}+w_h\|h_l(\boldsymbol{x},\boldsymbol{u})+\nabla h_l(\boldsymbol{x},\boldsymbol{u}).\delta\boldsymbol{x}\|_1+\dots\tag{6.89}$$

$$\dots+w_g\sum_{j=1}^{m}\max\{0,g_j(\boldsymbol{x},\boldsymbol{u})+\nabla g_j(\boldsymbol{x},\boldsymbol{u}).\delta\boldsymbol{x}\}\tag{6.90}$$

The linearised cost function for the subproblem to be solved in every iteration is given below,

$$\text{LP3: minimise } :=f(\boldsymbol{x},\boldsymbol{u})+\nabla f(\boldsymbol{x},\boldsymbol{u}).\delta\boldsymbol{x}+w_\nu\sum_{l=1}^{n}\|\boldsymbol{\nu}_l\|_2+w_g\sum_{j=1}^{m}\boldsymbol{\zeta}_j\tag{6.91}$$

The purpose of penalisation for the linear inequality constraints is to make them inactive during the optimisation process. This is advantageous for the constraints, that are *soft*, i.e., they do not need to be fulfilled precisely. However, the disadvantage of this inactivation due to penalisation results in the solution being not exactly like that of the original problem. Another disadvantage is the increase in the penalty parameters strictly enforces the constraints making the problem ill-conditioned by using large gradients and abrupt

function changes (Kurt and Yosi (2005)). Hence, we will remove the penalisation of the constraints from the cost function. Thus the cost functions for the discretised problem take the form,

$$\begin{aligned}
&\text{P2: minimise } \boldsymbol{N}(\boldsymbol{x},\boldsymbol{u}) := f(\boldsymbol{x},\boldsymbol{u}) + w_h \|h_l(\boldsymbol{x},\boldsymbol{u})\|_1 \\
&\text{LP2: minimise } \boldsymbol{L}(\boldsymbol{x},\boldsymbol{u}) := f(\boldsymbol{x},\boldsymbol{u}) + \nabla f(\boldsymbol{x},\boldsymbol{u}).\delta\boldsymbol{x} + w_h \|h_l(\boldsymbol{x},\boldsymbol{u}) + \nabla h_l(\boldsymbol{x},\boldsymbol{u}).\delta\boldsymbol{x}\|_1 \\
&\text{P3: minimise } f(\boldsymbol{x},\boldsymbol{u}) + w_v \sum_{l=1}^{n} \|\boldsymbol{v}_l\|_2 \\
&\text{LP3: minimise } f(\boldsymbol{x},\boldsymbol{u}) + \nabla f(\boldsymbol{x},\boldsymbol{u}).\delta\boldsymbol{x} + w_v \sum_{l=1}^{n} \|\boldsymbol{v}_l\|_2
\end{aligned} \tag{6.92}$$

6.3.5. Virtual Control

Linearisation of the dynamics can lead to the introduction of infeasibility in the solution space even if the original non-linear problem is feasible. This situation is called *artificial infeasibility*. For example, if a function, $f_1(x) = 2x^4$ is linearised to first order about $x = 1$, $f_2(x) = 8x - 6$. As can be seen in Figure 6.9, certain states become unreachable and the stationary point at $x = 0$ becomes infeasible. This undesirable infeasiblity prevents convergence as it obstructs the iteration process.

Mao et al. (2016) suggest the use of a *virtual control*, $\boldsymbol{v}(t) \in \mathbb{R}^m$, that works as an addition to the actual control input to make the infeasible region, feasible. An unconstrained virtual control ensures, that any state in the feasible region of the original problem is reachable in finite time when just the control input $\boldsymbol{u}(t)$ is not enough to do so. The virtual control can be inferred as a an artificial or synthetic acceleration, that acts on the SC, to steer it to any state in the feasible region. The first order linearised dynamics from discretisation, therefore, is relaxed to,

$$\boldsymbol{x}_{k+1} = \tilde{\boldsymbol{A}}_k \boldsymbol{x}_k + \tilde{\boldsymbol{B}}_k \boldsymbol{u}_k + \boldsymbol{E}_k \boldsymbol{v}_k + \tilde{\boldsymbol{z}}_k \tag{6.93}$$

where, $\boldsymbol{E}_k \in \mathbb{R}^{n\times n}$ can be chosen based on $\boldsymbol{A}_k$, such that both of them $(\boldsymbol{A}(.),\ \boldsymbol{E}(.))$ are controllable.

The virtual control is unconstrained, which means there is no restriction on its need, which needs to be taken care of, to ensure, that the control is still within the real bounds and can be achieved by the SC subject to the original non-linear constraints. This is done by *penalising* the cost function with an additional term of the form,

$$\gamma(.) := \operatorname*{ess\,sup}_{t\in[0,t_f]} \|.(t_k)\|_2 \tag{6.94}$$

where, $\|.\|_2$ is the Euclidean norm on $\mathbb{R}^n$ given by $\|c(t)\|_2 = \left(\sum_{i=1}^{n} c^2(t)\right)^{1/2}$. The virtual control is represented by $\boldsymbol{E}\boldsymbol{v}$ and hence the penalty function becomes,

$$\gamma(\boldsymbol{E}\boldsymbol{v}) := \operatorname*{ess\,sup}_{t\in[0,t_f]} \|\boldsymbol{E}_k \boldsymbol{v}_k\|_2 \tag{6.95}$$

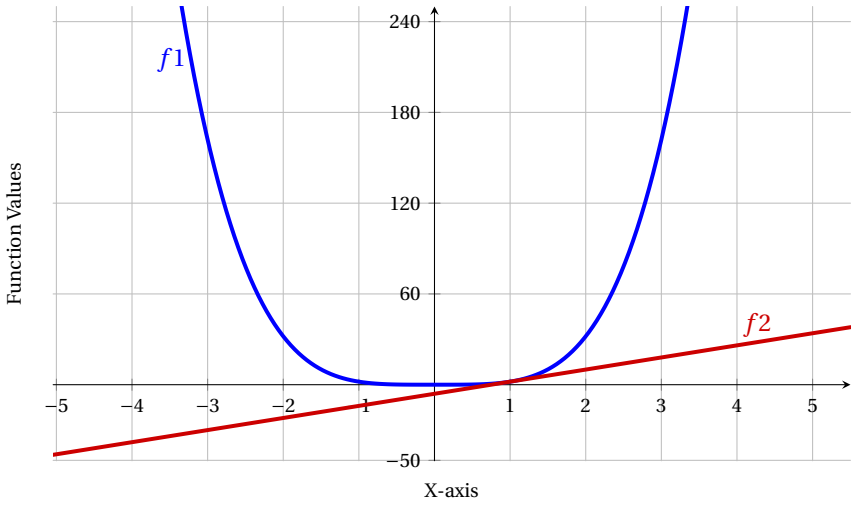


Figure 6.9: Plots of $f_1(x) = 2x^4$ and its first order linearised form $f_2(x) = 8x - 6$ at $x = 1$

The new penalised cost function becomes,

$$\text{minimise} - m(t_f) + w_v \gamma(\boldsymbol{E}\boldsymbol{v}) \tag{6.96}$$

where, w_v is the penalty weight for the virtual control. The magnitude of this weight is quite large to make sure that the deviation that is allowed by the virtual control during artificial infeasibility does not exist in regions where a feasible solution exists from the linearised dynamics.

As can be seen, the cost function is no longer linear and needs to be linearised for the problem formulation to be convex. This is done by introducing slack variables in twofold for the exact penalty function in the cost. The first step is to introduce a scalar slack variable, s_{v_k} to bound the virtual controls at every time step and also penalise the cost function for the same. With the penalty function in the form of the second norm of the slack variable, the cost function becomes,

$$\text{minimise} - m(t_f) + w_v \, \|s_{v_k}\|_2 \tag{6.97}$$

where, $\|.\|_1$ is the first norm on $\mathbb{R}^n$ given by $\|c(t)\|_1 = \sum_{i=1}^{n} |c(t)|$. The additional constraint on the virtual control becomes,

$$\|\boldsymbol{v}_k\|_2 \leq s_{v_k} \tag{6.98}$$

This can then be linearised by an additional slack variable, S_{v_k}, such that the cost function can be convex.

$$\text{minimise} - m(t_f) + w_v \, S_{v_k} \tag{6.99}$$

with a bound on the second norm of the slack variables, s_{v_k} over all the time steps,

$$\|s_{v_k}\|_2 \leq S_{v_k} \tag{6.100}$$

With these constraints the virtual control is bounded to prevent artificial infeasibility and the cost function is linearised to be handled by an SOCP solver.

6.3.6. Trust Region

The linearisation of non-linear kinematics and dynamics may cause the problem to become *unbounded*. This happens when the approximation of intermediate states allows a large deviation from the linearising point. The approximation is then rendered incapable of capturing the non-linearity. As can be seen, the non-linear problem will have a stationary point at $x = 0$ even if $x \in \mathbb{R}_-$ whereas the minimum of the linearised problem is unbounded and can achieve a minimum as the minimum negative x possible in the set. The graph in Figure 6.9 shows the two functions clearly showing how the linearised function becomes unbounded.

This unbounded nature of the linearised function can be restricted employing *trust regions* called the Trust Region Method (TRM). This region is a subset of the objective or cost function, which is modelled by means of a function, that is usually quadratic. The purpose of this is to define a region about the linearisation point, which is the current best solution, such that the best solution from linear approximation lies within it. Likewise, if it is within the trust region the subset is expanded or kept the same otherwise it is contracted. This ensures that the linearised trajectory does not diverge from the nominal trajectory attained in the previous succession by a significant amount to maintain numerical stability thereby preventing unboundedness.

For defining the quadratic function for the trust, we need to define the deviations involved. The deviation in the state is defined by, $\delta\boldsymbol{x} \triangleq \boldsymbol{x}_k - \bar{\boldsymbol{x}}_k$ and the deviation in control as, $\delta\boldsymbol{u} \triangleq \boldsymbol{u}_k - \bar{\boldsymbol{u}}_k$ (Szmuk et al. (2017)). Using these definitions the convex quadratic inequality for the trust region is expressed as,

$$\delta\boldsymbol{x}_k^T \delta\boldsymbol{x}_k + \delta\boldsymbol{u}_k^T \delta\boldsymbol{u}_k \leq \eta_k \tag{6.101}$$

where, $\eta(t)$ is a new variable called the *trust region radius* and $\eta \in \mathbb{R}^K$. The constraint is a quadratic constraint of the form shown below,

$$\boldsymbol{X}^{\mathrm{T}} A^{\mathrm{T}} A\boldsymbol{X} + b^{\mathrm{T}}\boldsymbol{X} + c \leq 0 \tag{6.102}$$

To convert this to SOCP, the equivalent constraint is[7],

$$\left\| \begin{matrix} (1 + b^{\mathrm{T}}\boldsymbol{X} + c)/2 \\ A\boldsymbol{X} \end{matrix} \right\|_2 \leq (1 - b^{\mathrm{T}}\boldsymbol{X} - c)/2 \tag{6.103}$$

[7] https://en.wikipedia.org/wiki/Second-order_cone_programming, date accessed:10-07-18

The trust region constraint can, therefore, be written in the SOCP format as,

$$\begin{aligned}(\boldsymbol{X}_{u,k}-\bar{\boldsymbol{X}}_{u,k})^{\mathrm{T}}(\boldsymbol{X}_{u,k}-\bar{\boldsymbol{X}}_{u,k}) &\leq \eta_k \\ \boldsymbol{X}_{u,k}^{\mathrm{T}}\boldsymbol{X}_{u,k}-2\bar{\boldsymbol{X}}_{u,k}^{\mathrm{T}}\boldsymbol{X}_{u,k}+\bar{\boldsymbol{X}}_{u,k}^{\mathrm{T}}\bar{\boldsymbol{X}}_{u,k} &\leq \eta_k \\ \left\| \begin{matrix} \left(1-2\bar{\boldsymbol{X}}_{u,k}^{\mathrm{T}}\boldsymbol{X}_{u,k}+(\bar{\boldsymbol{X}}_{u,k}^{\mathrm{T}}\bar{\boldsymbol{X}}_{u,k}-\eta_k)\right)/2 \\ \boldsymbol{I}_n\boldsymbol{X}_{u,k} \end{matrix} \right\|_2 &\leq \left(1+2\bar{\boldsymbol{X}}_{u,k}^{\mathrm{T}}\boldsymbol{X}_{u,k}-(\bar{\boldsymbol{X}}_{u,k}^{\mathrm{T}}\bar{\boldsymbol{X}}_{u,k}-\eta_k)\right)/2\end{aligned} \tag{6.104}$$

where, $\boldsymbol{X}_{u,k}$ is the unified state and control vector. The two main methods of implementing TRM are penalising the cost function for the trust region radius or implementing the trust region radius as a dynamic parameter based on the performance of the solution from $(i-1)^{th}$ iteration. Both the methods implement a hard constraint on the state solution and an ill defined η_k or logic for update of η_k would affect the convergence of the algorithm. A smaller value of η_k will lead to infeasibility for a poor initial trajectory whereas a large η_k would lead to bigger linearisation errors. The way to avoid these disadvantages is to use the trust region constraint as a soft constraint and reducing the sensitivity of the algorithm to η_k.

Wang and Cui (2018) in their work implement a dynamically changing the penalty weight for the trust region radius in the cost function and prove the improved performance for convergence in the successive convexification algorithm. We will go ahead with the same methodology as explained below. The cost function has to be penalised directly if the trust radius is a state variable. The additional term in the cost function due to the trust region is similar to the virtual control, we take the second norm of the trust radius.

$$\gamma(\eta) := \operatorname*{ess\,sup}_{t\in[0,t_f]} \|\eta_k\|_2 \tag{6.105}$$

and therefore, to bound the size of the trust region the non-linear cost function is directly penalised with a weighted (w_η) second norm of the trust region radius,

$$\text{minimise} - m(t_f) + w_\nu\, \gamma(\boldsymbol{E}\boldsymbol{\nu}) + w_\eta \gamma(\boldsymbol{\eta}) \tag{6.106}$$

Similar to virtual control, we introduce a slack variable, S_{η_k} to linearise the cost function,

$$\text{minimise} - m(t_f) + w_\nu\, S_{\nu_k} + w_\eta\, S_{\eta_k} \tag{6.107}$$

An additional constraint is, therefore, required on the trust radius,

$$\|\boldsymbol{\eta}_k\|_2 \leq S_{\eta_k} \tag{6.108}$$

For the dynamic implementation a performance criterion is required to analyse the quality of the solution obtained from $(i-1)^{th}$ iteration. This is done by comparing the linear and non-linear cost functions are compared at every iteration, i. A new ratio is introduced for this, denoted by ϱ^i, which compares the realised cost reduction, $\Delta\boldsymbol{N}^i$ to the predicted cost reduction, $\Delta\boldsymbol{L}^i$.

$$\begin{aligned}\Delta\boldsymbol{N}_c^i &= \boldsymbol{N}_c(\bar{\boldsymbol{x}}^i,\bar{\boldsymbol{u}}^i) - \boldsymbol{N}_c(\boldsymbol{x}^i,\boldsymbol{u}^i) \\ \Delta\boldsymbol{L}_c^i &= \boldsymbol{N}_c(\bar{\boldsymbol{x}}^i,\bar{\boldsymbol{u}}^i) - \boldsymbol{L}_c(\boldsymbol{x}^i-\bar{\boldsymbol{x}}^i,\boldsymbol{u}^i-\bar{\boldsymbol{u}}^i) \\ \varrho^i &= \frac{\Delta\boldsymbol{N}_c^i}{\Delta\boldsymbol{L}_c^i}\end{aligned} \tag{6.109}$$

where, the value of ϱ^i, is evaluated based on four parameters, that enable the trust radius update for the next iteration. These parameters follow as, $0<\varrho_0<\varrho_1<\varrho_2$. The resulting cases from them are given as below,

Case 1: $\varrho^i \leq \varrho_0 \ll 1$
This case essentially means that predicted cost reduction overestimates the actual reduction in cost and hence the linear approximation is inaccurate. This case requires the iteration to be rejected and the trust radius to be contracted. This contraction of the trust radius ensures that this case occurs a finite number of times rather than the algorithm getting stuck at the particular iteration. The penalty has to be, therefore, increased by a factor of σ and redo the iteration i.

Case 2: $\varrho_0 \leq \varrho^i < \varrho_1$
This case suggests, that the linearisation is deficient in its accuracy, but can still be accepted, but with a lower trust region (increasing penalty by a factor of $\sigma<1$) for the next iteration $i+1$ while accepting the solution $(\boldsymbol{x}^i,\boldsymbol{u}^i)$. This is done, so that the accuracy of the next iteration is better while avoiding the discarding of the previous iteration.

Case 3: $\varrho_1 \leq \varrho^i < \varrho_2$
This case occurs when the linearisation is sufficiently accurate, such that the predicted and actual cost reduction are very close, but not the same. This proves that the trust radius is a sufficient constraint and the penalty is kept the same for the next iteration $i+1$ while keeping the solution $(\boldsymbol{x}^i, \boldsymbol{u}^i)$.

Case 4: $\varrho_2 \leq \varrho^i$
When the linearisation is an exact or conservative approximation of the non-linear problem, this case occurs, which means the prediction is either equivalent to or underestimates the actual reduction. The trust region can, therefore, be enlarged for the next iteration by reducing the penalty by a factor of ς and the previous iterate solution is accepted.

All the cases have been represented below,

$$
\begin{aligned}
w_\eta^i &= \sigma\, w_\eta^i, \quad \text{if } \varrho^i \leq \varrho_0 \ll 1 \\
w_\eta^{i+1} &= \begin{cases} \sigma\, w_\eta^i, & \text{if } \varrho_0 \leq \varrho^i < \varrho_1 \\ w_\eta^i, & \text{if } \varrho_1 \leq \varrho^i < \varrho_2 \\ w_\eta^i/\varsigma, & \text{if } \varrho_2 \leq \varrho^i \end{cases}
\end{aligned} \tag{6.110}
$$

Note: In the case, that $\Delta \boldsymbol{L}^i = 0$, the solution can be accepted and there is no need to compute the ratio. Since, this means, that the non-linear and linear costs are exactly the same.

6.3.7. SCvx Problem Formulation

In this section, we reformulate the problem in the relative frame using DQs as shown in Problem 1E to that using discretised DQ dynamics and kinematics and SOCP constraints with the penalised cost functions to be solved for every iteration, i.

From the earlier sections on virtual control and trust region, the unified state will contain the state, the controls, the virtual controls, its slack variables and the trust region radius and its slack variable. Since, SCvx involves linearisation and discretisation of the dynamics, the need for slack variables for the non-convex mass dynamics and thrust constraint becomes unnecessary, and the problem formulation can return to its original form with the addition of the virtual control and trust region radius and their slack variables. The problem formulation is shown on the below in Problem 1F.

➽ PROBLEM 1F: Problem Formulation for SCvx

minimise $-m_{t_f} + w_\nu\, S_{\nu_k} + w_\eta\, S_{\eta_k}$

subject to:

Kinematics and Dynamics:
$\boldsymbol{x}_{k+1} = \tilde{\boldsymbol{A}}_k \boldsymbol{x}_k + \tilde{\boldsymbol{B}}_k \boldsymbol{u}_k + \boldsymbol{E}_k \boldsymbol{\nu}_k + \tilde{\boldsymbol{z}}_k$

Control Constraints:
$-m_{t_f} \leq -m_k, \quad m_k \leq m_{t_0}$
$F_{min} \leq f_{\boldsymbol{F}^{\mathrm{B}},k} \leq F_{max}$
$\|\boldsymbol{\nu}_k\|_2 \leq s_{\nu_k}$
$\|s_{\nu_k}\|_2 \leq S_{\nu_k}$

Trust Region Constraints:
$$\left\| \begin{matrix} \left(1 - 2\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}} \boldsymbol{X}_{u,k} + (\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}} \bar{\boldsymbol{X}}_{u,k} - \eta_k)\right)/2 \\ \boldsymbol{I}_n \boldsymbol{X}_{u,k} \end{matrix} \right\|_2 \leq \left(1 + 2\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}} \boldsymbol{X}_{u,k} - (\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}} \bar{\boldsymbol{X}}_{u,k} - \eta_k)\right)/2$$
$\|\boldsymbol{\eta}_k\|_2 \leq S_{\eta_k}$

Boundary Conditions:
$m_0 = m_{wet}, \quad \boldsymbol{\omega}_{\mathtt{A/I}}^{\mathtt{A}} = \text{constant}$
$\dot{\boldsymbol{q}}_{\mathtt{B/A}}(0) = \dot{\boldsymbol{q}}_{\mathtt{B/A}_0}, \quad \dot{\boldsymbol{q}}_{\mathtt{B/A}}(t_f) = \dot{\boldsymbol{q}}_{\mathtt{B/A}_0}, \quad \dot{\boldsymbol{\omega}}_{\mathtt{B/A}}^{\mathtt{B}}(0) = \dot{\boldsymbol{\omega}}_{\mathtt{B/A}_0}^{\mathtt{B}}, \quad \dot{\boldsymbol{\omega}}_{\mathtt{B/A}}^{\mathtt{B}}(t_f) = \dot{\boldsymbol{\omega}}_{\mathtt{B/A}_f}^{\mathtt{B}}$

Additional Constraints:
$\|A_j(\boldsymbol{x}) + b_j\|_2 \leq c_j^T \boldsymbol{x} + d_j, \quad j = 1, \ldots, m$

6.3.8. SCvx Algorithm

The SCvx algorithm is as follows,
Initialisation:

1. for $k \in [0, K]$:

$$
\begin{aligned}
\bar{m}_k &= \frac{K-k}{K} m_{wet} + \frac{k}{K} m_{dry} \\
\check{\boldsymbol{q}}_{\mathtt{B/A}_k} &= ScLERP\,(\frac{k}{K};\ \check{\boldsymbol{q}}_{\mathtt{B/A}_0},\ \check{\boldsymbol{q}}_{\mathtt{B/A}_f}) \\
\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}_k} &= \frac{K-k}{K} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}_0} + \frac{k}{K} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}_f} \\
\check{\boldsymbol{F}}^{\mathtt{B}}_k &= [-\bar{m}_k \boldsymbol{g}^{\mathtt{B}}_k,\ 0,\ r_{F^{\mathtt{B}}} \times (-\bar{m}_k \boldsymbol{g}^{\mathtt{B}}_k),\ 0]^{\mathtt{T}} \\
\dot{\check{\boldsymbol{F}}}^{\mathtt{B}}_k &= \boldsymbol{0} \\
\bar{\boldsymbol{x}}^0_k &= \left[\bar{m}_k,\ \check{\boldsymbol{q}}_{\mathtt{B/A}_k},\ \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}_k},\ \check{\boldsymbol{F}}^{\mathtt{B}}_k.\ \dot{\check{\boldsymbol{F}}}^{\mathtt{B}}_k\right] \\
\bar{\boldsymbol{u}}^0_k &= \boldsymbol{0}
\end{aligned}
\tag{6.111}
$$

2. for $k \in [0, K)$: Compute $\dot{\bar{\boldsymbol{x}}}^0_k,\ \tilde{\boldsymbol{A}}_k,\ \tilde{\boldsymbol{B}}_k,\ \tilde{\boldsymbol{z}}_k$

SCvx Loop:
for $i \in [1, i_{max})$

1. Use $\bar{\boldsymbol{x}}^{i-1}_k,\ \bar{\boldsymbol{u}}^{i-1}_k,\ \tilde{\boldsymbol{A}}^{i-1}_k,\ \tilde{\boldsymbol{B}}^{i-1}_k,\ \tilde{\boldsymbol{z}}^{i-1}_k$ to solve the subproblem LP3 and obtain $\bar{\boldsymbol{x}}^i_k,\ \bar{\boldsymbol{u}}^i_k$.
2. Compute ϱ^i and update w^i_η
3. Set $\boldsymbol{x}^i_k \rightarrow \bar{\boldsymbol{x}}^i_k$ and $\boldsymbol{u}^i_k \rightarrow \bar{\boldsymbol{u}}^i_k$.
4. Compute $\tilde{\boldsymbol{A}}^i_k,\ \tilde{\boldsymbol{B}}^i_k,\ \tilde{\boldsymbol{z}}^i_k$ and move to next iteration.
5. If $\|\eta^i\|_2 \leq tol$ **exit loop** else move to **next iteration**.

6.3.9. Stacking Equations

The SOCP solvers are not built to deal with second norms in the cost function, and hence we need to introduce new variables to involve the virtual control and trust region in the concatenated state matrix for the solver. The OCP format, that ECOS handles is as follows[8],

$$
\begin{aligned}
&\text{minimise } \boldsymbol{c}^{\mathbb{T}} \boldsymbol{x} \\
&\quad \text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\
&\qquad \boldsymbol{G}_1 \boldsymbol{x} \preceq_K \boldsymbol{h}_1 \\
&\qquad \boldsymbol{G}_2 \boldsymbol{x} \leq \boldsymbol{h}_2
\end{aligned}
\tag{6.112}
$$

where the symbol $\preceq_K$ denotes a generalised inequality with respect to the cone K, i.e., the vector $s = h - \boldsymbol{G}\boldsymbol{x} \in K$. The stacked equations consist of the state and control vectors along with the virtual controls and trust region radii, we denote the concatenated state vector for ECOS (Domahidi et al. (2013)) as,

$$\boldsymbol{X} = [\boldsymbol{x}_0\ \boldsymbol{x}_1\ \boldsymbol{x}_3\ \ldots \boldsymbol{x}_K]^{\mathtt{T}} \tag{6.113}$$

$$\boldsymbol{U} = [\boldsymbol{u}_0\ \boldsymbol{u}_1\ \boldsymbol{u}_3\ \ldots \boldsymbol{u}_{K-1}]^{\mathtt{T}} \tag{6.114}$$

$$\boldsymbol{N} = [\boldsymbol{\nu}_0\ \boldsymbol{\nu}_1\ \boldsymbol{\nu}_3\ \ldots \boldsymbol{\nu}_{K-1}]^{\mathtt{T}} \tag{6.115}$$

$$\boldsymbol{Z} = [\tilde{\boldsymbol{z}}_0\ \tilde{\boldsymbol{z}}_1\ \tilde{\boldsymbol{z}}_3\ \ldots \tilde{\boldsymbol{z}}_{K-1}]^{\mathtt{T}} \tag{6.116}$$

[8] https://www.embotech.com/ECOS, date accessed:10-08-18

The stacked up equations for the equality constraints representing the dynamics have been put forth. For the first order solution,

$$\begin{bmatrix} \boldsymbol{I}_n & & & & & & \\ -\tilde{\boldsymbol{A}}_0 & \boldsymbol{I}_n & & & & & \\ \boldsymbol{0} & -\tilde{\boldsymbol{A}}_1 & \boldsymbol{I}_n & & & & \\ \boldsymbol{0} & \boldsymbol{0} & -\tilde{\boldsymbol{A}}_2 & \boldsymbol{I}_n & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\tilde{\boldsymbol{A}}_3 & \boldsymbol{I}_n & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ \boldsymbol{0} & & & \dots & & -\tilde{\boldsymbol{A}}_{K-1} & \boldsymbol{I}_n \end{bmatrix} \begin{pmatrix} \boldsymbol{x}(0) \\ \boldsymbol{x}(1) \\ \boldsymbol{x}(2) \\ \boldsymbol{x}(3) \\ \vdots \\ \boldsymbol{x}(K) \end{pmatrix} = \begin{bmatrix} \boldsymbol{0} & & & & & \\ \tilde{\boldsymbol{B}}_0 & & & & & \\ \boldsymbol{0} & \tilde{\boldsymbol{B}}_1 & & & & \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{B}}_2 & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{B}}_3 & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{0} & & \dots & & & \tilde{\boldsymbol{B}}_{K-1} \end{bmatrix} \begin{pmatrix} \boldsymbol{u}(0) \\ \boldsymbol{u}(1) \\ \boldsymbol{u}(2) \\ \boldsymbol{u}(3) \\ \vdots \\ \boldsymbol{u}(K-1) \end{pmatrix} + \\ + \begin{bmatrix} \boldsymbol{0} & & & & & \\ \boldsymbol{I}_n & & & & & \\ \boldsymbol{0} & \boldsymbol{I}_n & & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{0} & & \dots & & & \boldsymbol{I}_n \end{bmatrix} \begin{pmatrix} \boldsymbol{\nu}(0) \\ \boldsymbol{\nu}(1) \\ \boldsymbol{\nu}(2) \\ \boldsymbol{\nu}(3) \\ \vdots \\ \boldsymbol{\nu}(K-1) \end{pmatrix} + \begin{bmatrix} \boldsymbol{0} & & & & & \\ \boldsymbol{I}_n & & & & & \\ \boldsymbol{0} & \boldsymbol{I}_n & & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{0} & & \dots & & & \boldsymbol{I}_n \end{bmatrix} \begin{pmatrix} \tilde{\boldsymbol{z}}(0) \\ \tilde{\boldsymbol{z}}(1) \\ \tilde{\boldsymbol{z}}(2) \\ \tilde{\boldsymbol{z}}(3) \\ \vdots \\ \tilde{\boldsymbol{z}}(K-1) \end{pmatrix} \tag{6.117}$$

Similarly, the higher order solution is represented below and its stacking for ECOS is done in same fashion as the first order case.

$$\begin{bmatrix} \boldsymbol{I}_n & & & & & & \\ -\tilde{\boldsymbol{A}}_0 & \boldsymbol{I}_n & & & & & \\ \boldsymbol{0} & -\tilde{\boldsymbol{A}}_1 & \boldsymbol{I}_n & & & & \\ \boldsymbol{0} & \boldsymbol{0} & -\tilde{\boldsymbol{A}}_2 & \boldsymbol{I}_n & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\tilde{\boldsymbol{A}}_3 & \boldsymbol{I}_n & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ \boldsymbol{0} & & & \dots & & -\tilde{\boldsymbol{A}}_{K-1} & \boldsymbol{I}_n \end{bmatrix} \begin{pmatrix} \boldsymbol{x}(0) \\ \boldsymbol{x}(1) \\ \boldsymbol{x}(2) \\ \boldsymbol{x}(3) \\ \vdots \\ \boldsymbol{x}(K) \end{pmatrix}$$

$$= \begin{bmatrix} \boldsymbol{0} & & & & & \\ \tilde{\boldsymbol{B}}_0 & & & & & \\ \tilde{\boldsymbol{A}}_1\tilde{\boldsymbol{B}}_0 & \tilde{\boldsymbol{B}}_1 & & & & \\ \tilde{\boldsymbol{A}}_2\tilde{\boldsymbol{A}}_1\tilde{\boldsymbol{B}}_0 & \tilde{\boldsymbol{A}}_2\tilde{\boldsymbol{B}}_1 & \tilde{\boldsymbol{B}}_2 & & & \\ \tilde{\boldsymbol{A}}_3\tilde{\boldsymbol{A}}_2\tilde{\boldsymbol{A}}_1\tilde{\boldsymbol{B}}_0 & \tilde{\boldsymbol{A}}_3\tilde{\boldsymbol{A}}_2\tilde{\boldsymbol{B}}_1 & \tilde{\boldsymbol{A}}_3\tilde{\boldsymbol{B}}_2 & \tilde{\boldsymbol{B}}_3 & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ \tilde{\boldsymbol{A}}_{K-1}\tilde{\boldsymbol{A}}_{K-2}\dots\tilde{\boldsymbol{A}}_1\tilde{\boldsymbol{B}}_{\boldsymbol{0}} & & \dots & & & \tilde{\boldsymbol{B}}_{K-1} \end{bmatrix} \begin{pmatrix} \boldsymbol{u}(0) \\ \boldsymbol{u}(1) \\ \boldsymbol{u}(2) \\ \boldsymbol{u}(3) \\ \vdots \\ \boldsymbol{u}(K-1) \end{pmatrix} + \dots$$

$$\dots + \begin{bmatrix} \boldsymbol{0} & & & & & \\ \boldsymbol{I}_n & & & & & \\ \boldsymbol{0} & \boldsymbol{I}_n & & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{0} & & \dots & & & \boldsymbol{I}_n \end{bmatrix} \begin{pmatrix} \boldsymbol{\nu}(0) \\ \boldsymbol{\nu}(1) \\ \boldsymbol{\nu}(2) \\ \boldsymbol{\nu}(3) \\ \vdots \\ \boldsymbol{\nu}(K-1) \end{pmatrix} + \begin{bmatrix} \boldsymbol{0} & & & & & \\ \boldsymbol{I}_n & & & & & \\ \tilde{\boldsymbol{A}}_1 & \boldsymbol{I}_n & & & & \\ \tilde{\boldsymbol{A}}_2\tilde{\boldsymbol{A}}_1 & \tilde{\boldsymbol{A}}_2 & \boldsymbol{I}_n & & & \\ \tilde{\boldsymbol{A}}_3\tilde{\boldsymbol{A}}_2\tilde{\boldsymbol{A}}_1 & \tilde{\boldsymbol{A}}_3\tilde{\boldsymbol{A}}_2 & \tilde{\boldsymbol{A}}_3 & \boldsymbol{I}_n & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ \tilde{\boldsymbol{A}}_{K-1}\tilde{\boldsymbol{A}}_{K-2}\dots\tilde{\boldsymbol{A}}_1 & & \dots & & & \boldsymbol{I}_n \end{bmatrix} \begin{pmatrix} \tilde{\boldsymbol{z}}(0) \\ \tilde{\boldsymbol{z}}(1) \\ \tilde{\boldsymbol{z}}(2) \\ \tilde{\boldsymbol{z}}(3) \\ \vdots \\ \tilde{\boldsymbol{z}}(K-1) \end{pmatrix} \tag{6.118}$$

Therefore, the stacked dynamic equations take the compact form as,

$$\boldsymbol{Q}_x\boldsymbol{X} = \boldsymbol{Q}_u\boldsymbol{U} + \boldsymbol{Q}_\nu\boldsymbol{N} + \boldsymbol{Q}_z\boldsymbol{Z} \tag{6.119}$$

where, n is 33 for each time step, k the DQ state vector, $\boldsymbol{X}$ is the stacked state solution vector, $\boldsymbol{Q}_x$ is the stacked system matrix, $\boldsymbol{U}$ is the stacked control vector, $\boldsymbol{Q}_u$ is the stacked input matrix, $\boldsymbol{N}$ is the stacked virtual control vector, $\boldsymbol{Q}_\nu$ is the stacked virtual input matrix, $\boldsymbol{Z}$ is the stacked dynamics solution state vector, $\boldsymbol{Q}_z$ is the stacked matrix for the dynamics at the solution state. The zeros are of the dimensions $\boldsymbol{0}_{n\times n}$ in $\boldsymbol{Q}_x$, $\boldsymbol{0}_{n\times 8}$ in $\boldsymbol{Q}_u$, $\boldsymbol{0}_{n\times(n+1)}$ in $\boldsymbol{Q}_\nu$ and $\boldsymbol{0}_{n\times n}$ in $\boldsymbol{Q}_z$. The dimensions are, $\boldsymbol{X} \in \mathbb{R}^{((K+1)n)\times 1}$, $\boldsymbol{Q}_x \in \mathbb{R}^{((K+1)n)\times((K+1)n)}$, $(\boldsymbol{U}, \boldsymbol{Z}, \boldsymbol{N}) \in \mathbb{R}^{((K-1)n)\times 1}$ and $(\boldsymbol{Q}_u, \boldsymbol{Q}_z, \boldsymbol{Q}_\nu) \in \mathbb{R}^{((K+1)n)\times((K-1)n)}$.

The stacked equations for the inequality constraints need to extract parts of the lumped state and controls at every time step and constrain them. We deal with two kinds of constraints, either linear or SOCP. Beginning with the first linear constraint, which is for the mass of SC constraint as discussed in Problem 1F.

$$-m_{t_f} \leq -m_k \tag{6.120}$$

$$m_k \leq m_{t_0} \tag{6.121}$$

The extraction matrix for any time step can be written as,

$$-m_{t_f} \leq -\begin{bmatrix}\mathbf{0}_{1\times k(2n+10)} & 1 & \mathbf{0}_{1\times(K-k)(2n+10)-1}\end{bmatrix}\boldsymbol{X}_l \tag{6.122}$$

$$\begin{bmatrix}\mathbf{0}_{k\times(2n+10)} & 1 & \mathbf{0}_{(K-k)\times(2n+10)-1}\end{bmatrix}\boldsymbol{X}_l \leq m_{t_0} \tag{6.123}$$

where $k \in 0,..,K$, $\boldsymbol{X}_l$ is the lumped state, controls, virtual controls and trust region radius vector. The extraction matrix can be shorthanded as $\boldsymbol{G}_{k,m}$, such that it varies with every time step, k and is of the dimension $\mathbb{R}^{1\times K(2n+10)}$ and there are K such constraints.

$$-m_{t_f} \leq -\boldsymbol{G}_{k,m}\boldsymbol{X}_l \tag{6.124}$$

$$\boldsymbol{G}_{k,m}\boldsymbol{X}_l \leq m_{t_0} \tag{6.125}$$

Following the same for the lower bound on the thrust exerted by the propulsion engine,

$$\left(F_{min} - \left\|\bar{\boldsymbol{F}}_k^{\mathsf{B}}\right\|\right)\left(\frac{\bar{\boldsymbol{F}}^{\mathsf{B}}(t)}{\left\|\bar{\boldsymbol{F}}_k^{\mathsf{B}}\right\|}\right)^{-1} - \boldsymbol{F}^{\mathsf{B}}(t) \leq \boldsymbol{G}_{k,F}\boldsymbol{X}_l \tag{6.126}$$

where, $\boldsymbol{G}_{k,F} \in \mathbb{R}^{3\times K(2n+10)}$ and can be given as below,

$$\boldsymbol{G}_{k,F} = \begin{bmatrix}\mathbf{0}_{3\times k(2n+10)-(n-16)} & \boldsymbol{I}_3 & \mathbf{0}_{3\times(K-k)(2n+10)-3}\end{bmatrix} \tag{6.127}$$

These are the only linear constraints we deal with for the descent problem and the rest are all SOCP constraints. We begin with the upper bound on the thrust.

$$\left\|\boldsymbol{F}_k^{\mathsf{B}}\right\| \leq F_{max} \tag{6.128}$$

which can be written as below using the same extraction matrix as Eq. (6.127),

$$\left\|\boldsymbol{G}_{k,F}\boldsymbol{X}_l\right\| \leq F_{max} \tag{6.129}$$

In a similar fashion, we find the extraction matrices for the virtual controls and its slack variables and the slack variables for the trust region radius and write them in the below format,

$$\left\|\boldsymbol{G}_{k,v}\boldsymbol{X}_l\right\| \leq \boldsymbol{G}_{k,s_v}\boldsymbol{X}_l \tag{6.130}$$

$$\left\|\boldsymbol{G}_{k,s_v}\boldsymbol{X}_l\right\| \leq \boldsymbol{G}_{k,S_v}\boldsymbol{X}_l \tag{6.131}$$

$$\left\|\boldsymbol{G}_{k,\eta}\boldsymbol{X}_l\right\| \leq \boldsymbol{G}_{k,S_\eta}\boldsymbol{X}_l \tag{6.132}$$

$$\tag{6.133}$$

The trust region radius has to be converted from quadratic to SOCP as discussed in Eq. (6.104),

$$\left\|\begin{matrix}\left(1-2\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}}\boldsymbol{X}_{u,k}+(\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}}\bar{\boldsymbol{X}}_{u,k}-\eta_k)\right)/2 \\ \boldsymbol{I}_n\boldsymbol{X}_{u,k}\end{matrix}\right\|_2 \leq \left(1+2\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}}\boldsymbol{X}_{u,k}-(\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}}\bar{\boldsymbol{X}}_{u,k}-\eta_k)\right)/2 \tag{6.134}$$

This is broken down to two SOCP constraints of the format $\|A_k(\boldsymbol{x})+b_k\| \leq c_k^T\boldsymbol{x}+d_k$ in a single one,

$$b_k = \left(1+\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}}\bar{\boldsymbol{X}}_{u,k}\right)/2 \tag{6.135}$$

$$A_k = \begin{bmatrix}\mathbf{0}_{1\times k(n+8)} & -\bar{\boldsymbol{X}}_{u,k} & \mathbf{0}_{1\times(n+1)} & -1 & \mathbf{0}_{1\times(K-k-1)(n+8)-(n+2)} \\ \mathbf{0}_{(n+8)\times k(n+8)} & \boldsymbol{I}_{(n+8)} & \mathbf{0}_{(n+8)\times(n+1)} & 1 & \mathbf{0}_{(n+8)\times(K-k-1)(n+8)-(n+2)}\end{bmatrix} \tag{6.136}$$

$$c_k = \begin{bmatrix}\mathbf{0}_{1\times k(n+8)} & \bar{\boldsymbol{X}}_{u,k} & \mathbf{0}_{1\times(n+2)} & \mathbf{0}_{1\times(K-k-1)(n+8)-(n+2)}\end{bmatrix} \tag{6.137}$$

$$d_k = \left(1-\bar{\boldsymbol{X}}_{u,k}^{\mathsf{T}}\bar{\boldsymbol{X}}_{u,k}\right)/2 \tag{6.138}$$

Finally the there has to be an equality constraint on the final state of the trajectory, that enforces the pose and attitude of the SC and the thrust, but not the final mass. This can simply be achieved as below,

$$\begin{bmatrix}\mathbf{0}_{1\times(K-1)(2n+10)+1} & \mathbf{1}_{1\times(n-1)}\end{bmatrix}\boldsymbol{X}_{l_K} = \begin{bmatrix}\check{\boldsymbol{q}}_{\mathtt{B}/\mathtt{A}_f} & \check{\boldsymbol{\omega}}_{\mathtt{B}/\mathtt{A}_f} & \check{\boldsymbol{F}}_f^B & \dot{\check{\boldsymbol{F}}}_f^B\end{bmatrix}^{\mathtt{T}} \tag{6.139}$$

and is added to the linearised dynamics determined in Eq. (6.119).

The cost function also has to be converted, such that an extraction vector picks out the final mass and penalises the trust region and virtual controls as shown in Problem 1F. This is to satisfy the requirement, that an SOCP problem must be subject to an affine cost, as shown in Eq. (6.112). The cost function, therefore, can be written as,

$$\begin{pmatrix}\mathbf{0}_{(K-1)(2n+10)\times 1}\\ 1\\ \mathbf{0}_{((2n+10)-1)\times 1}\\ w_\nu\\ w_\eta\end{pmatrix}^{\mathtt{T}}\boldsymbol{X}_l = \boldsymbol{c}^{\mathtt{T}}\boldsymbol{X}_l \tag{6.140}$$

This completes the conversion of the descent problem to a format ECOS can understand and we use it to implement our problem in MATLAB® with the ECOS solver. The verification of this implementation can be found in Chapter 7.

6.4. Extended Convex Guidance

The powered descent guidance algorithm can prove to be challenging, because it usually treats the translational and rotational dynamics of the SC separately. This makes it difficult to deal with situations where both these dynamics need to be combined for some mission requirement. Certain constraints like the one on the line of sight of the SC require the SC to keep the target in sight while descent, which makes it dependent on both the orientation and position of the SC. In this section, we will parameterise all such constraints, that need to be met while the powered descent phase is in action. The additional constraints for a safe and optimal descent, have been derived for the first time using a dual quaternion state in the relative frame.

6.4.1. Need for Extended Constraints

We have the following requirements for the safety of the SC:

- The SC needs to avoid hazards like boulders or rough or highly sloped terrains near the landing site or those, that may not have been observed during mapping. (Section 6.4.2)
- The SC needs to follow a trajectory, such that no part of the SC comes in contact with the target surface before the end of the flight. (Section 6.4.2)
- The SC needs to maintain a certain attitude to make sure the landing site is within a certain angle of the line of sight of the navigation camera and the RADAR. (Section 6.4.3)
- The angular rate of the SC cannot exceed beyond a certain maximum allowable angular rate, for observational and safety purposes. (Section 6.4.3)
- The SC can exert the propulsive thrust in directions within the allowable gimbal of the engine, and hence a constraint needs to be imposed on the same. (Section 6.4.4)
- The SC needs to maintain a certain altitude at the end of the flight so that just the sample collector hardware is in contact with the surface, which also requires the enforcing of no subterranean flight to prevent any kind of damage to the hardware or SC. This is given by, defining the boundary condition on the final SC position, as, $\boldsymbol{r}_f = \boldsymbol{r}_{LM}^{\mathtt{A}} + r_{arm}\cdot\boldsymbol{e}_{LM}^{\mathtt{A}}$, where $\boldsymbol{r}_{LM}^{\mathtt{A}}$ and $\boldsymbol{e}_{LM}^{\mathtt{A}}$ are the position vector of the landing site and the normal from the landing site in the relative frame.

6.4.2. Glide-Slope Constraint

The guidance problem for the powered descent is a TPBVP, which means it finds an optimal trajectory from the initial state of the SC to a required final state or landing site. The control constraints discussed in the previous section take care of the feasibility of the trajectory from the point of view of the propulsion system. However, we need to implement certain path constraints, since this is a TAG descent.

In lieu of all the above requirements, the *glide-slope constraint* can be established, which enforces on the optimiser software a safe reachable space for the SC. The glide slope is nothing but a cone about the normal from the surface of the landing site with its vertex at the landing site location on the surface, which guides the trajectory of the SC to ensure no contact with the surface before the end of the flight. In other words, this means, that the elevation angle of the SC with respect to the plane of the surface must always be greater than the glide-slope angle, γ. Geometrically this is represented in the Figure 6.10 and mathematically can be represented as,

$$\left\| \boldsymbol{r}^{\mathtt{A}}_{SC} - \boldsymbol{r}^{\mathtt{A}}_{LM} \right\| \cos\gamma \leq (\boldsymbol{r}^{\mathtt{A}}_{SC} - \boldsymbol{r}^{\mathtt{A}}_{LM}) \cdot \boldsymbol{e}^{\mathtt{A}}_{LM} \quad \forall\, t \in [t_0, t_f] \tag{6.141}$$

where, $\boldsymbol{r}^{\mathtt{A}}_{SC}$, $\boldsymbol{r}^{\mathtt{A}}_{LM}$ are the position vectors of the SC and landing site respectively and $\boldsymbol{e}^{\mathtt{A}}_{LM}$ is the normal from the landing site in the asteroid rotating reference frame, $\mathtt{A}$. This takes care of the second and third requirements mentioned above. Now with the HDA system in the loop, retargeting might be required on the detection of hazards. One such trigger could be the presence of an unobserved boulder[9] near the landing site. This can be taken care of by simply increasing the glide-slope angle by the amount required as shown in Figure 6.11 and resolving the optimisation problem. The same treatment is implemented for a surface with an increasing slope close to the landing site.

$$\theta_b = \arctan\left(\frac{h_b + \delta h_b}{\sqrt{x_b^2 + y_b^2}} \right) \tag{6.142}$$

where, θ_b is the new glide-slope angle for safe approach of the SC, h_b is the height of the boulder, δh_b is an additional safety height above the boulder and x_b, y_b are the X and Y co-ordinates of the position of the boulder in the $\mathtt{A}$ frame obtained from $\boldsymbol{r}^{\mathtt{A}}_{LM} - \boldsymbol{r}^{\mathtt{A}}_{boulder}$.
The inability of this constraint to safeguard the SC arises in situations where the HDA software detects one or more hazards at the landing the site for example rocky terrain or an undetected crater (could be the rim or slope or pit). In this case, a new trajectory to a nearby safe landing site needs to be solved for. The implementation of the logic of optimised landing on a new safe site or retargeting will be left for future work.

The glide-slope constraint from Eq. (6.141) needs to be redefined in terms of DQs. We first take the LHS,

$$(\boldsymbol{r}^{\mathtt{A}}_{SC} - \boldsymbol{r}^{\mathtt{A}}_{LM}) \cdot \boldsymbol{e}^{\mathtt{A}}_{LM} = \boldsymbol{r}^{\mathtt{A}}_{SC} \cdot \boldsymbol{e}^{\mathtt{A}}_{LM} - \boldsymbol{r}^{\mathtt{A}}_{LM} \cdot \boldsymbol{e}^{\mathtt{A}}_{LM}$$

The first term in the expansion can be rewritten with the help of the unit quaternion triple identity (can be

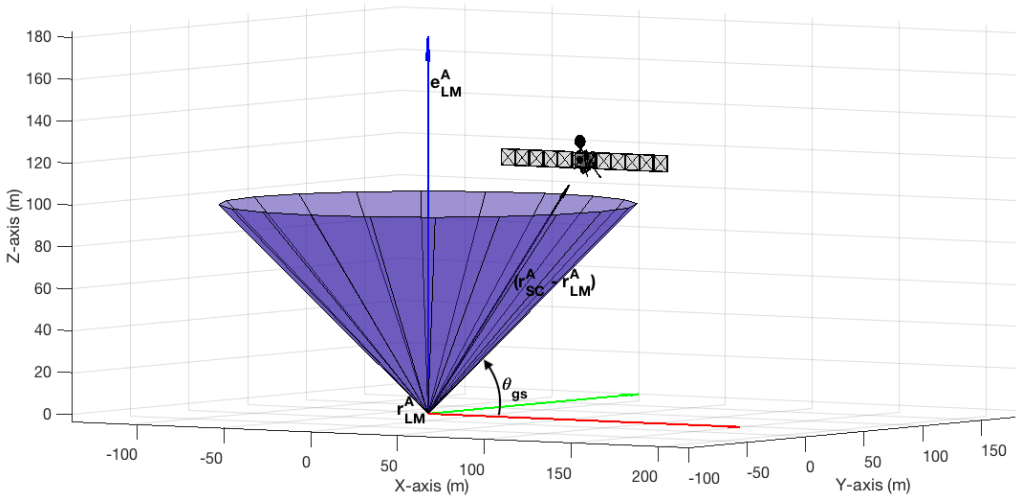


Figure 6.10: Physical representation of glide-slope constraint.

[9]Mapping will detect the tall boulders, only boulders of small height need to be considered. Implementing the presence of tall boulders would make the safe glide slope angle too large and make the landing scenario infeasible.

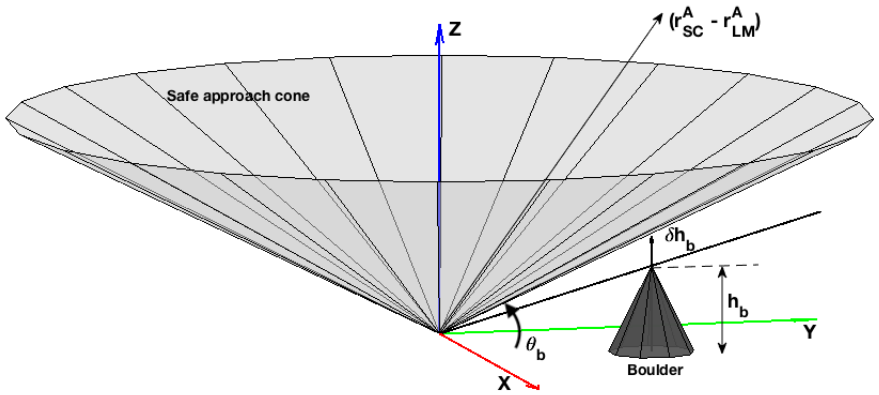


Figure 6.11: Physical representation of glide-slope constraint.

found in Appendix C). The proof for this identity can be found in Lee and Mesbahi (2015).

$$\begin{aligned} \boldsymbol{r}^{\mathtt{A}}_{SC} \cdot \boldsymbol{e}^{\mathtt{A}}_{LM} &= (\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC})^{\mathtt{T}} (\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{e}^{\mathtt{A}}_{LM}) \\ &= (\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC})^{\mathtt{T}} \left[\boldsymbol{e}^{\mathtt{A}}_{LM}\odot\right] \boldsymbol{q}_{\mathtt{B/A}} \\ &= \begin{pmatrix} \boldsymbol{q}_{\mathtt{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC} \end{pmatrix}^{\mathtt{T}} \begin{bmatrix} \boldsymbol{0}_{4\times4} & \left[\boldsymbol{e}^{\mathtt{A}}_{LM}\odot\right]^{\mathtt{T}} \\ \left[\boldsymbol{e}^{\mathtt{A}}_{LM}\odot\right] & \boldsymbol{0}_{4\times4} \end{bmatrix} \begin{pmatrix} \boldsymbol{q}_{\mathtt{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC} \end{pmatrix} \\ &= \check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_1} \check{\boldsymbol{q}}_{\mathtt{B/A}} \end{aligned} \tag{6.143}$$

Since, $\boldsymbol{r}^{\mathtt{A}}_{LM}$ and $\boldsymbol{e}^{\mathtt{A}}_{LM}$ are always constant $(\forall\, t)$ in the asteroid frame of reference, $\mathtt{A}$, their dot product is also a constant and will be represented as c_{GS}. The LHS, therefore, becomes,

$$(\boldsymbol{r}^{\mathtt{A}}_{SC} - \boldsymbol{r}^{\mathtt{A}}_{LM}) \cdot \boldsymbol{e}^{\mathtt{A}}_{LM} = \check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_1} \check{\boldsymbol{q}}_{\mathtt{B/A}} - c_{GS} \tag{6.144}$$

Now for the RHS of Eq. (6.141),

$$\left\|\boldsymbol{r}^{\mathtt{A}}_{SC} - \boldsymbol{r}^{\mathtt{A}}_{LM}\right\| \cos\gamma = \left\|\boldsymbol{r}^{\mathtt{A}}_{SC} \cdot \boldsymbol{r}^{\mathtt{A}}_{SC} - 2\boldsymbol{r}^{\mathtt{A}}_{SC} \cdot \boldsymbol{r}^{\mathtt{A}}_{LM} + \boldsymbol{r}^{\mathtt{A}}_{LM} \cdot \boldsymbol{r}^{\mathtt{A}}_{LM}\right\| \cos\gamma$$

Using the unit quaternion triple identity matrix for the first term in the above expanded norm,

$$\begin{aligned} \boldsymbol{r}^{\mathtt{A}}_{SC} \cdot \boldsymbol{r}^{\mathtt{A}}_{SC} &= (\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC})^{\mathtt{T}} (\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC}) \\ &= \begin{pmatrix} \boldsymbol{q}_{\mathtt{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC} \end{pmatrix}^{\mathtt{T}} \begin{bmatrix} \boldsymbol{0}_{4\times4} & \boldsymbol{0}_{4\times4} \\ \boldsymbol{0}_{4\times4} & 4\boldsymbol{I}_4 \end{bmatrix} \begin{pmatrix} \boldsymbol{q}_{\mathtt{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC} \end{pmatrix} \\ &= \check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_3} \check{\boldsymbol{q}}_{\mathtt{B/A}} \end{aligned} \tag{6.145}$$

Similarly for the second term in the expanded norm,

$$\begin{aligned} \boldsymbol{r}^{\mathtt{A}}_{SC} \cdot \boldsymbol{r}^{\mathtt{A}}_{LM} &= (\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC})^{\mathtt{T}} (\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{LM}) \\ &= \begin{pmatrix} \boldsymbol{q}_{\mathtt{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC} \end{pmatrix}^{\mathtt{T}} \begin{bmatrix} \boldsymbol{0}_{4\times4} & \left[\boldsymbol{r}^{\mathtt{A}}_{LM}\odot\right]^{\mathtt{T}} \\ \left[\boldsymbol{r}^{\mathtt{A}}_{LM}\odot\right] & \boldsymbol{0}_{4\times4} \end{bmatrix} \begin{pmatrix} \boldsymbol{q}_{\mathtt{B/A}} \\ \frac{1}{2}\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{r}^{\mathtt{A}}_{SC} \end{pmatrix} \\ &= \check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_4} \check{\boldsymbol{q}}_{\mathtt{B/A}} \end{aligned} \tag{6.146}$$

and the last term inside the expanded norm being a constant is represented by b_{GS}, which gives the form of RHS as,

$$\begin{aligned} \left\|\boldsymbol{r}^{\mathtt{A}}_{SC} - \boldsymbol{r}^{\mathtt{A}}_{LM}\right\| \cos\gamma &= \left\|\check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_3} \check{\boldsymbol{q}}_{\mathtt{B/A}} - 2\check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_4} \check{\boldsymbol{q}}_{\mathtt{B/A}} + b_{GS}\right\| \cos\gamma \\ &= \left\|\check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_2} \check{\boldsymbol{q}}_{\mathtt{B/A}} + b_{GS}\right\| \cos\gamma \end{aligned} \tag{6.147}$$

where, $\boldsymbol{M}_{GS_2} = \boldsymbol{M}_{GS_3} - 2\boldsymbol{M}_{GS_4}$. The glide-slope constraint using DQs in the relative frame can now be represented as,

$$-\check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_1} \check{\boldsymbol{q}}_{\mathtt{B/A}} + \left\|\check{\boldsymbol{q}}^{\mathtt{T}}_{\mathtt{B/A}} \boldsymbol{M}_{GS_2} \check{\boldsymbol{q}}_{\mathtt{B/A}} + b_{GS}\right\| \cos\gamma + c_{GS} \leq 0 \tag{6.148}$$

With closer inspection, it can be found, that the matrices, $\boldsymbol{M}_{GS_1}$ and $\boldsymbol{M}_{GS_2}$ are *indefinite symmetric* and *indefinite* respectively. The nature of these matrices obstructs the conversion of the DQ glide-slope constraint to the SOCP form. To convert a quadratic constraint to SOCP, the quadratic term in it must have a *semi-definite positive* matrix, such that it can be decomposed into the form $L^{\mathrm{T}}L$ as shown for the trust region constraint. Review of available methods for decomposition of indefinite matrices like Bunch-Kaufman and Cholesky decomposition methods show a possible decomposition to $L^{\mathrm{T}}DL$, which is undesirable for the required conversion to SOCP. The only available method to make the glide-slope constraint convex and suitable for the SOCP formulation is to linearise it. This will be left for future work.

6.4.3. Attitude Constraints

The rotational manoeuvres of the SC need to be restricted depending on the requirement of the onboard instruments. For altitude measurements for example via RADAR the attitude of the SC should be within a certain angle with respect to the normal to the surface being observed for accurate measurements as well as within a certain angle with respect to the boresight of the RADAR or optical camera due to sensor hardware limitations.

LOS Constraint

The LOS constraint is about the body-fixed frame. The visual representation of the LOS constraint is shown in the Figure 6.12. The mathematical interpretation of the constraint as shown below,

$$\left\|\boldsymbol{r}_{SC}^{\mathtt{B}}-\boldsymbol{r}_{LM}^{\mathtt{B}}\right\|\cos\iota\leq-(\boldsymbol{r}_{SC}^{\mathtt{B}}-\boldsymbol{r}_{LM}^{\mathtt{B}})\cdot\boldsymbol{e}_{cam}^{\mathtt{B}}\quad\forall\,t\in[t_0,t_f] \tag{6.149}$$

where, $\boldsymbol{e}_{cam}^{\mathtt{B}}$ is a unit vector along the boresight of the NavCam or LIDAR onboard the SC in its body frame,$\mathtt{B}$. Treating the LOS constraint similar to the glide-slope constraint to get its formulation in DQ, the LOS constraint becomes,

$$\check{\boldsymbol{q}}_{\mathtt{B}/\mathtt{A}}^{\mathrm{T}}\boldsymbol{M}_{LOS_1}\check{\boldsymbol{q}}_{\mathtt{B}/\mathtt{A}}+\left\|\check{\boldsymbol{q}}_{\mathtt{B}/\mathtt{A}}^{\mathrm{T}}\boldsymbol{M}_{LOS_2}\check{\boldsymbol{q}}_{\mathtt{B}/\mathtt{A}}+b_{LOS}\right\|\cos\iota-c_{LOS}\leq0 \tag{6.150}$$

where, b_{LOS} and c_{LOS} are no more constants and change with every time step with the quaternion $\boldsymbol{q}_{B/A}$. They are given as,

$$b_{LOS}=(\boldsymbol{q}_{B/A}\otimes\boldsymbol{r}_{LM}^{\mathtt{A}}\otimes\boldsymbol{q}_{B/A}^{*})^{\mathrm{T}}(\boldsymbol{q}_{B/A}\otimes\boldsymbol{r}_{LM}^{\mathtt{A}}\otimes\boldsymbol{q}_{B/A}^{*}) \tag{6.151}$$

$$c_{LOS}=(\boldsymbol{q}_{B/A}\otimes\boldsymbol{r}_{LM}^{\mathtt{A}}\otimes\boldsymbol{q}_{B/A}^{*})^{\mathrm{T}}\boldsymbol{e}_{cam}^{\mathtt{B}} \tag{6.152}$$

The matrices $\boldsymbol{M}_{LOS_1}$ and $\boldsymbol{M}_{LOS_2}$ are given by,

$$\boldsymbol{M}_{LOS_1}=\begin{bmatrix}\boldsymbol{0}_{4\times4} & \left[\boldsymbol{e}_{cam}^{\mathtt{B}}\otimes\right]^{\mathrm{T}}\\ \left[\boldsymbol{e}_{cam}^{\mathtt{B}}\otimes\right] & \boldsymbol{0}_{4\times4}\end{bmatrix} \tag{6.153}$$

$$\boldsymbol{M}_{LOS_2}=\begin{bmatrix}\boldsymbol{0}_{4\times4} & -\left[\boldsymbol{r}_{LM}^{\mathtt{B}}\otimes\right]^{\mathrm{T}}\\ -\left[\boldsymbol{r}_{LM}^{\mathtt{B}}\otimes\right] & 4\boldsymbol{I}_4\end{bmatrix} \tag{6.154}$$

As can be seen the LOS constraint has additional complications as compared to the glide-slope constraint and the best way to overcome this constraint is to linearise it as well. Since, the addition of these complex constraints are a add on to the actual algorithm and make the solution more realistic, they can be worked on a later phase after the SCvx algorithm is functional. Hence, the linearisation and implementation of the LOS constraint is left for future work as well. For this thesis maintaining the tilt angle of the SC within a certain limit will suffice for the attitude to not be unrealistic.

Tilt Angle Constraint

Now, the constraint for accurate measurements is an angle about the surface normal(ground fixed frame) whereas

$$\boldsymbol{e}_{LM_z}^{\mathtt{A}}\cdot\boldsymbol{e}_{SC_x}^{\mathtt{A}}\geq\cos\delta\quad\forall\,t\in[t_0,t_f] \tag{6.155}$$

Simplifying the RHS we get,

$$\begin{aligned}\boldsymbol{e}_{LM_z}^{\mathtt{A}}\cdot\boldsymbol{e}_{SC_x}^{\mathtt{A}}&=\boldsymbol{e}_{LM_z}^{\mathtt{A}}\cdot\left(\boldsymbol{C}_{\mathtt{A}/\mathtt{B}}\boldsymbol{e}_{SC_x}^{\mathtt{B}}\right)\\&=\boldsymbol{e}_{LM_z}^{\mathtt{A}}\cdot\left(1-2\left[q_{\mathtt{B}/\mathtt{A}_2}^2+q_{\mathtt{B}/\mathtt{A}_3}^2\right]\right)\end{aligned} \tag{6.156}$$

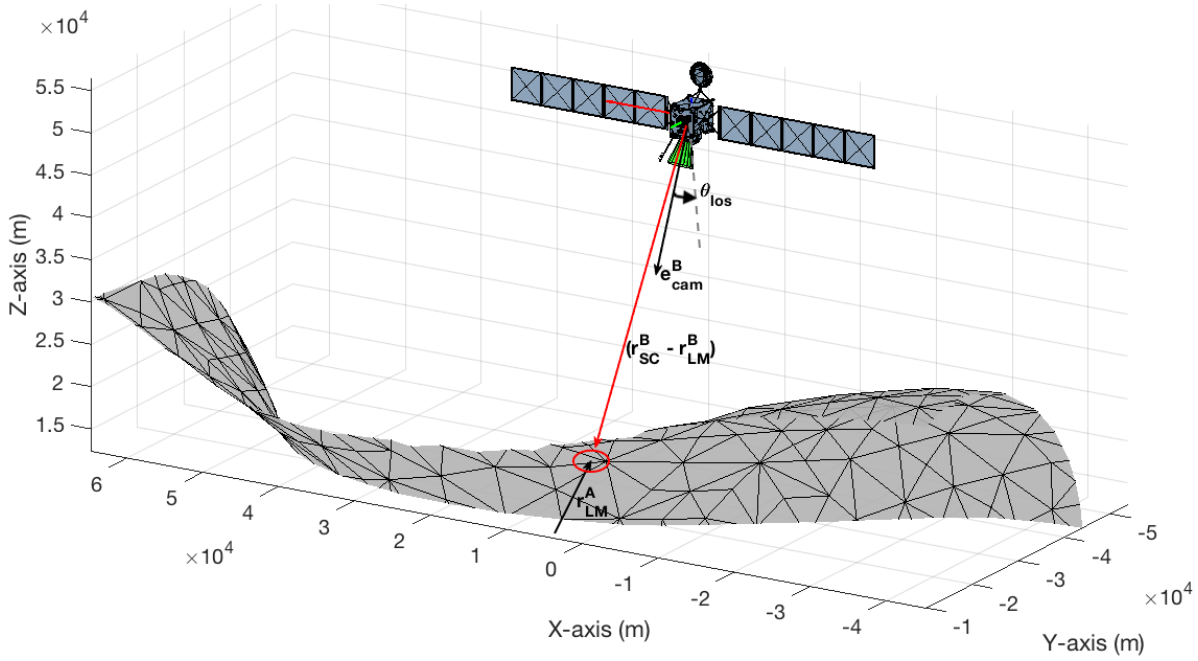


Figure 6.12: Physical representation of LOS constraint (SC not to scale).

The constraint can be rewritten as,

$$\left\| \boldsymbol{e}^{\mathtt{A}}_{LM_z} \cdot \left(1 - 2\left[q^2_{\mathtt{B}/\mathtt{A}_2} + q^2_{\mathtt{B}/\mathtt{A}_3}\right]\right)\right\|_2 \leq \cos\delta \tag{6.157}$$

The extraction matrix of the constraint looks like,

$$\boldsymbol{G}_{k,\theta} = \begin{bmatrix} \boldsymbol{0}_{3\times k(2n+10)-(n-24)} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times(K-k)(2n+10)-3} \end{bmatrix} \tag{6.158}$$

and the final discrete tilt angle constraint looks like,

$$\left\| \boldsymbol{G}_{k,\theta}\boldsymbol{X}_l \right\| \leq \cos_\delta \tag{6.159}$$

Angular Rate Constraint

The third attitude constraint is one on the angular rate of the SC. This constraint is to maintain a rate below a maximum allowable value. It becomes an inherent SOCP constraint as shown below,

$$\left\| \tilde{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B}/\mathtt{A}_k} \right\|_2 \leq \omega_{max} \tag{6.160}$$

Since, we do constrain the translational velocity, the extraction matrix for this constraint becomes,

$$\boldsymbol{G}_{k,\omega} = \begin{bmatrix} \boldsymbol{0}_{3\times k(2n+10)-(n-24)} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times(K-k)(2n+10)-3} \end{bmatrix} \tag{6.161}$$

and the constraint can now be written as,

$$\left\| \boldsymbol{G}_{k,\omega}\boldsymbol{X}_l \right\| \leq \omega_{max} \tag{6.162}$$

6.4.4. Thrust Direction Constraint

Thrust pointing/direction constraint as the name suggests is the constraint in the body-fixed frame for the direction of thrust from the propulsion system. The importance of this constraint is that it couples the feasibility of the physical trajectory to the SC hardware capabilities of the propulsion system. In case of a rigidly fixed thruster system to the SC body, this constraint is the same as the attitude constraint, since the direction of the commanded thrust will determine the attitude. For the mission scenario of TAG descent, a gimballed thruster would prove to be more beneficial. This is, because the SC has a sample collector in the extended position whilst thrusting to successfully hover over the surface in the final phase of descent, which would

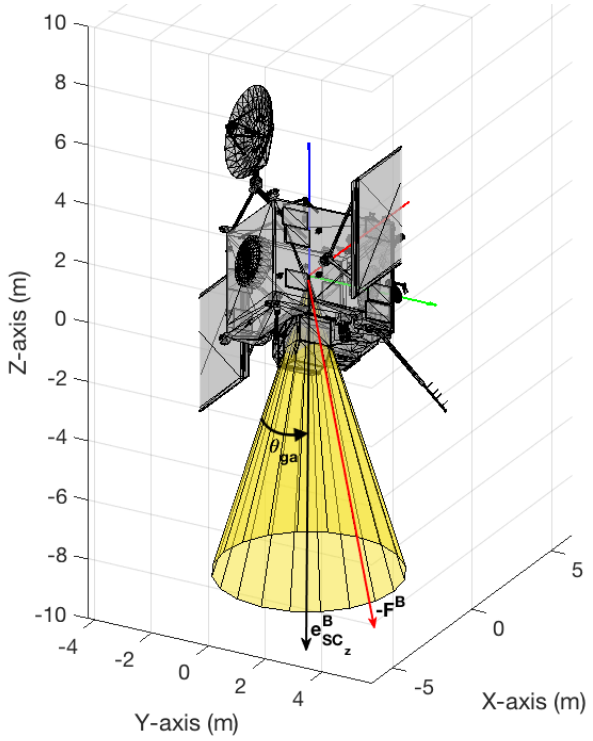


Figure 6.13: Physical representation of gimbal angle constraint.

require the thruster to avoid disturbing the surface from where the sample needs to be collected. The thrust direction constraint can be given as,

$$\left\| \boldsymbol{F}^{\mathrm{B}} \right\| \cos\theta \leq \boldsymbol{e}^{\mathrm{B}}_{SC_z} \cdot \boldsymbol{F}^{\mathrm{B}} \quad \forall\, t \in [t_0, t_f] \tag{6.163}$$

where, $\boldsymbol{e}^{\mathrm{B}}_{SC_z}$ is the direction in the SC body frame and $\cos\theta \leq \pi/2$ has to be about the direction to implement a convex conic constraint. The nonconvexity for $\cos\theta \geq \pi/2$ can be removed by using a slack variable for the thrust norm. A more detailed explanation can be found in Açıkmeşe and Ploen (2007). For now, it is enough to construct this continuous time constraint to discrete time with extraction matrices, as discussed in the previous section.

If the SC is treated as a point mass and a single thrust vector represents the net thrust, then the net thrust on the SC could be pointed arbitrarily. With a rigid body SC, an arbitrary net thrust is not possible, since the propulsion thruster cannot have any possible orientation with respect to the SC structure. The restrictions come from the SC design as well as the safety concern for onboard instruments. This can be translated to the thrust direction constraint where the net thrust vector would lie within a cone about the gimbal axis of the propulsion system. This can be seen in Figure 6.13. Converting Eq. (6.163) to one, that be handled by ECOS, we get,

$$\left\| \boldsymbol{G}_{k,F} \boldsymbol{X}_l \right\| \leq \frac{\boldsymbol{e}^{\mathrm{B}}_{SC_z}}{\cos\theta} \boldsymbol{G}_{k,F} \boldsymbol{X}_l \tag{6.164}$$

which is the last constraint formulation for the descent problem.

6.5. Conclusion

All the concepts required for the method of successive convexification have been documented in this chapter. The linearised kinematics and dynamics for a dual quaternion state in the relative frame have been derived. All the constraints required for a safe descent have been presented and derived in the dual quaternion format. With this chapter, the author has formulated successive convexification for a 6DOF descent guidance problem. To summarise the chapter, Figure 6.14, provides with the logical flow followed through it and can be used by the reader to avail the sections of interest.

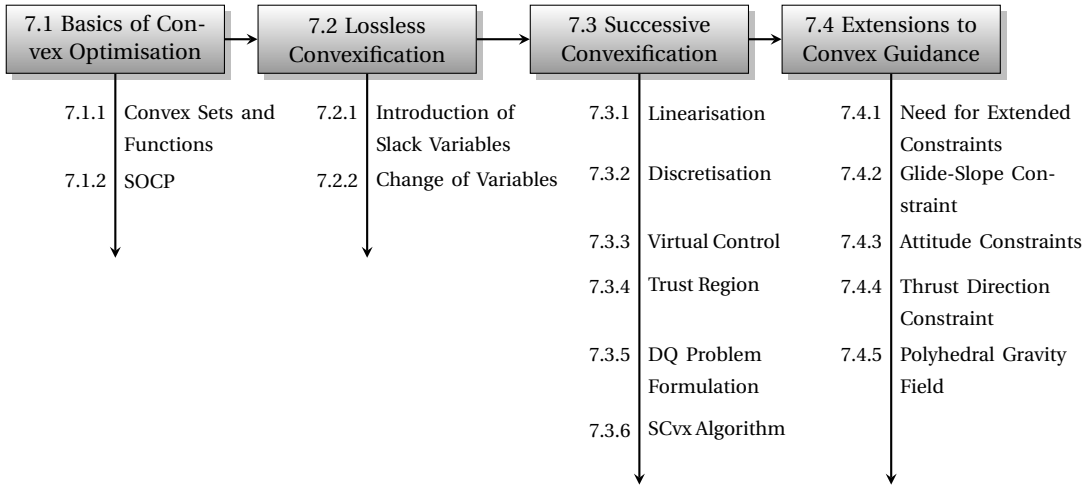


Figure 6.14: Chapter Overview

7

Simulator Design and Verification

The implementation of all the theoretical concepts discussed till now, requires developing a simulator that emulates the actual physical conditions of the mission scenario. Since the need for hardware in the loop is not present for developing a guidance algorithm, a simulator will serve the purpose of experimental setup for this thesis. To make sure this setup is well documented for further use and no string is left untied, we use this chapter from outlining the basic design of the simulator to all the verification tests performed to ensure its full and correct functionality.

7.1. Software Architecture

The simulation environment for this thesis has quite a few complicated implementations and therefore needs to be briefly introduced for better understanding. The architecture and some important details for the simulator is discussed in this section.

7.1.1. Top Level Architecture

A description of the functionality of the required blocks in a top level architecture for the guidance simulator has been provided below. They have been implemented in Figure 7.1 as required for the descent phase scenario. It gives an idea of the required data flow in the closed-loop system and can be used for the mapping phase as well with minor changes.

Inputs

- Algorithm Parameters: the boundary conditions implied in the problem formulations, the parameters that need to predefined like trajectory nodes, constraint switches and.
- SC Model: structure, reflectivity, mass properties, propulsion system parameters, initial conditions.
- Asteroid Model: polyhedral structure, polyhedral gravity, pre-generated landmark map, illumination map.
- Environment Model: external disturbances.
- Planetary Ephemerides: sun data, asteroid data.

Outputs

- Trajectory Data: solution state with complete 6DOF states for every node.
- Update Parameters: updated mass of SC, scaling factors for penalty weights.
- Simulator Performance: estimated accuracy, algorithm runtime, etc.

Post Processing

- processing of the data obtained from the simulator to be presented in a graphical or visual format for easy understanding.

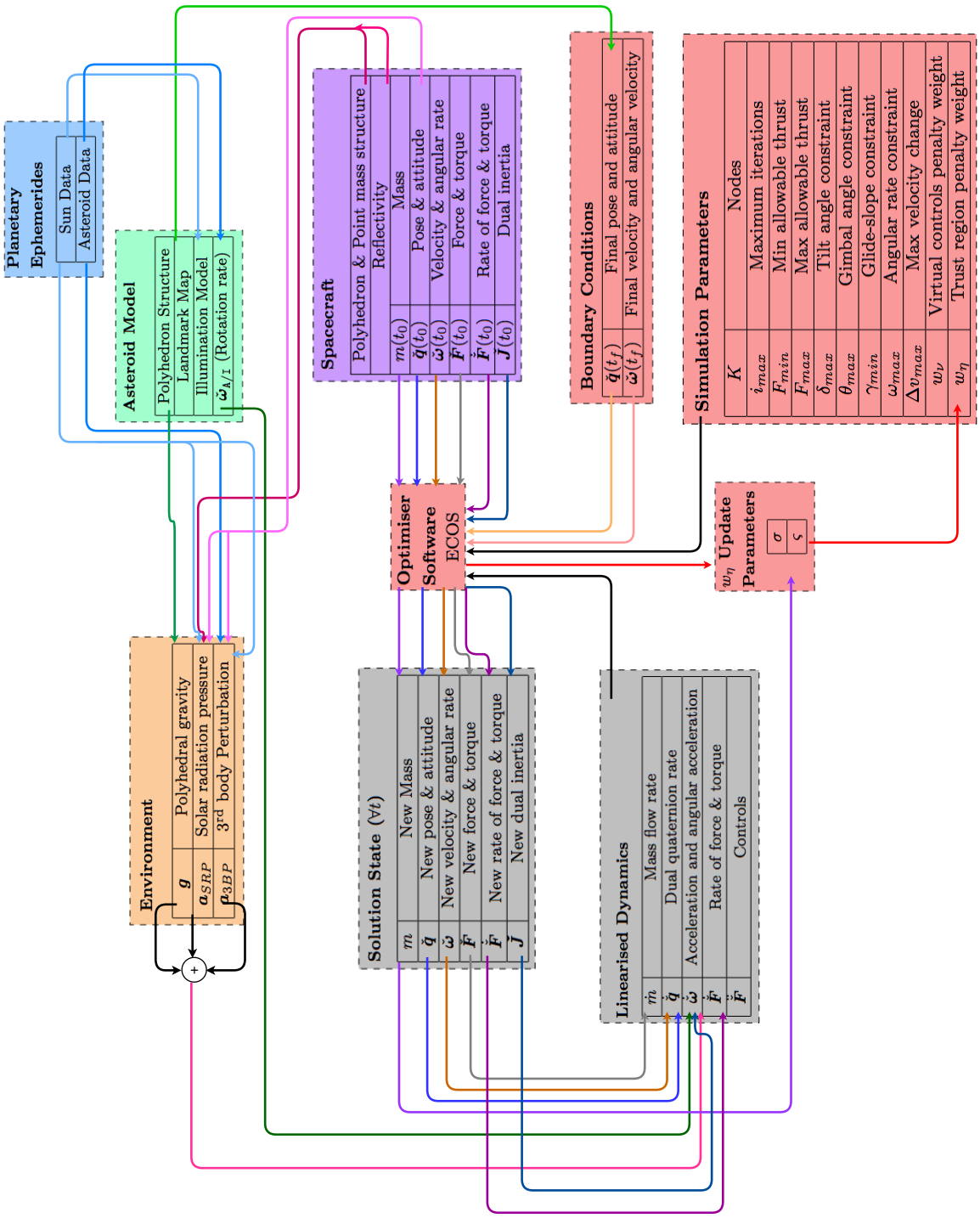


Figure 7.1: Simulator Architecture for Descent Guidance

7.1.2. Simulator Design

As mentioned in the top level architecture the simulations block is divided into one that takes care of the mapping phase of the asteroid and the other that takes care of the descent phase and they are discussed in detailed in the next subsections with a further breakdown of the top level software architecture. For both the mission phases, MATLAB® has been chosen for the computation for the following reasons:

- Its ease of use, availability and general familiarity with it.
- Convex solvers for MATLAB® are readily available. It has been proved to work well for the convex optimisation implementation of the 3DOF problem of descent Sagliano et al. (2017), Gerth (2014) and Ridder (2016).
- The algorithms being in development phase, there isn't a need for computations fast enough for real-time implementation. (It can be coded in C++ or converted to .mex files when needed for faster computations)
- Comparison with other open source softwares (FreeMat, Scilab, R, and IDL) was done by Ecaterina et al. (2012). The test results led to the conclusion that this software could perform with equivalent efficiency and/or accuracy as MATLAB® only for certain case of problems each and therefore MATLAB® is the best option.

Since the mission phases have to implement different guidance algorithms, it is important to break the simulation structure with respect to each and it can be found in Figures 7.2 and 7.3. They represent the flowcharts for the respective algorithms and also the softwares and sections where the theoretical concepts have been developed.

Mapping (Figure 7.2)

- Compute Dynamics: Based on the conditions at the initial state, the mesh of possible ΔV is used to compute the dynamics of the system.
- Propagate Trajectories: With the help of the dynamics the trajectories are propagated with a RK45 integrator whilst making sure that the obstacle space set is respected during each propagation.
- Solve Problem: With the trajectories propagated, their quality is examined by means of the functions discussed in Section A.5, and the most optimal trajectory is selected.
- Update and Refine Problem: Firstly, the mission completion status is updated with the found optimal trajectory. Secondly, the end state of the found optimal trajectory is now the initial state and the mesh for ΔV is refined towards the most interesting region. The process is reiterated from the 'Compute dynamics block', till the mission completion status is a 100%.

Descent (Figure 7.3)

- Linearise and Discretise: The solution states are obtained at every node and the continuous dynamic system is linearised and discretised to initialise the problem.
- Transcribe Problem: The problem is then transcribed to a format that can be read by ECOS to solve the problem. There are tools like 'CVX' that can be used to do this, but we do it in MATLAB® my manual transcription. It has been made such that, the user does not need to tweak the complex matrix structures for their specific states and controls, besides updating the input for the number of variables to be used.
- Solve Problem: The transcribed problem is fed to the Embedded Conic Optimisation Solver (ECOS) and the discrete time solution is found.

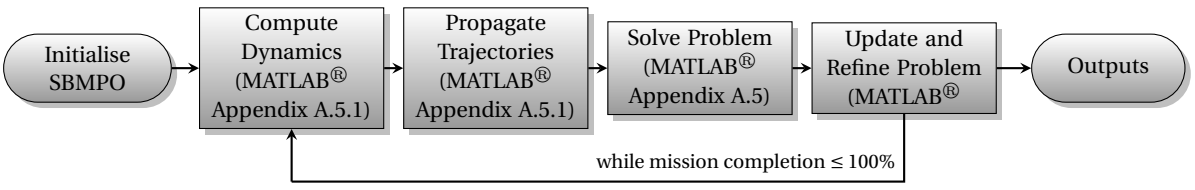


Figure 7.2: Detailed simulation block for SBMPO

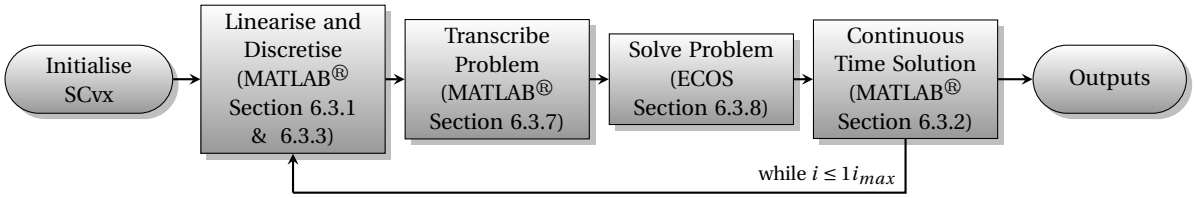


Figure 7.3: Detailed simulation block for SCvx Optimisation

- Continuous Time Solution: The discrete time state space solution is then converted to a continuous time command as discussed in Section 6.3.2.

7.1.3. Optimisation Solver

As discussed in Section 6.1, Boyd and Vandenberghe (2010) mention that the method of convex optimisation has almost reached the level of a mature technology. Many convex solvers have already been developed and are in use. Developing a new solver is out of scope for this thesis and we select ECOS, which is one of the solvers that are available to be implemented in the MATLAB® environment.

ECOS is a second order cone programming (SOCP) interior-point solver. It was designed for embedded applications and is written in low footprint, single threaded, library free ANSI-C. The interior point algorithm is based on the standard Mehrotra PECE method with Nesterov-Todd scaling and self-dual embedding (Sagliano et al. (2017)). For allowing stable factorisation with a fixed pivoting order, its search directions are found via a symmetric indefinite Karush-Kuhn-Tucker system.

The code for ECOS is open source and can be downloaded from https://github.com/embotech/ecos-matlab. It can be used with numerous interfaces like CVX, YALMIP, Python, Julia, Ansi-C, etc.[1] Gerth (2014) has verified that it is the fastest solver available and since it is specialised in handling SOCPs it is the best choice for this thesis.

The solver is not built to deal with second norms in the cost function or the form of stacked equations discussed in Section 6.3.9 and hence we need to redefine the stacked state matrix, system matrix and introduce new variables to involve the virtual control and trust region in the total state matrix for the solver. The OCP format that ECOS handles is as follows[2]:

$$\begin{aligned} &\text{minimise } \boldsymbol{c}^{\mathbb{T}}\boldsymbol{X} \\ &\text{s.t. } \boldsymbol{A}\boldsymbol{X} = \boldsymbol{b} \\ &\quad \boldsymbol{G}\boldsymbol{X} \preceq_K \boldsymbol{h} \end{aligned} \tag{7.1}$$

where the symbol $\preceq_K$ denotes a generalised inequality with respect to the cone K, i.e., the vector $d = h - \boldsymbol{G}\boldsymbol{x} \in K$. The stacked equations consist of the state and control vectors along with the virtual controls and trust region radii, we denote the concatenated state vector for ECOS as

$$\boldsymbol{X} = [\boldsymbol{x}_0\ \boldsymbol{u}_0\ \boldsymbol{\nu}_0\ \boldsymbol{s}_0\ \boldsymbol{\eta}_0\ \ldots \boldsymbol{x}_K\ \boldsymbol{u}_K\ \boldsymbol{\nu}_K\ \boldsymbol{s}_K\ \boldsymbol{\eta}_K\ S_\nu\ S_\eta]^{\mathrm{T}} \tag{7.2}$$

7.2. Spacecraft Model

In this section, the details for the SC model to be used in the simulator is detailed. The SC is based on Rosetta's dimensions, since the mission was to Comet 67P, which is at a distance of 3 au and hence it would satisfy for the power needs to similar distant missions. We discuss the dimensions in detail along with its mass and inertia properties and also its reflective properties.

7.2.1. Structure

The dimensions of Rosetta have been used as a reference for the SC model for the simulator. The general structure of the SC is that of a cuboid of dimensions $2 \times 2.1 \times 2.8$ m at the centre with two solar arrays of dimensions $14 \times 0 \times 2.28$ m at about 1 m away from the cuboid along the X-axis. As mentioned the width of the solar arrays is considered negligible. They have an area of 32 sq.m each. The total wingspan of the solar

[1] https://github.com/embotech/ecos-matlab, date accessed:17-08-18
[2] Same as 1

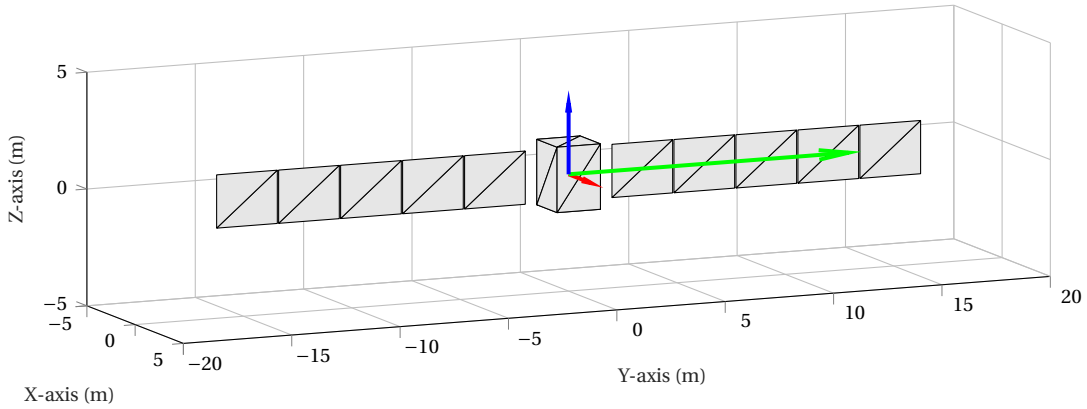


Figure 7.4: Simplified SC model based on Rosetta's dimensions for the mapping phase.

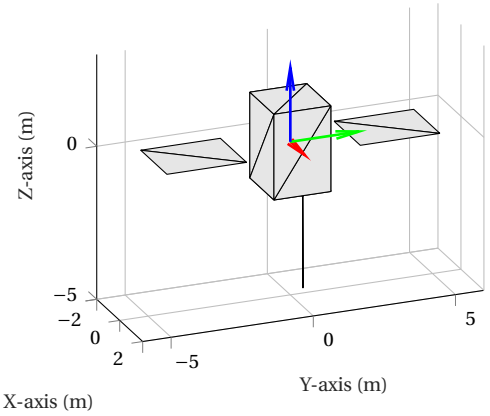


Figure 7.5: Simplified SC model based on steerable and foldable Rosetta panels and extended sample collection arm with dimensions equivalent to OSIRIS-REx for the descent phase.

arrays is about 32 m. For this thesis we use a simplified model of Rosetta, since the .stl file consists of more than ≈ 50000 vertices and this would take a toll on the computation time. Figure 7.4 shows the simplified SC model for the mapping phase.

For the descent phase, the SC is assumed to have folded solar panels. There solar arrays on Rosetta have five sub-panels on each solar array, as can be seen in Figure 7.4. They are folded such that, the panels reduce to the size of one sub-panel on each side of the SC block. The panels are then steered to have them face up, along the Z-axis. Also the sample collector mechanism is extended during this phase, which is 3 m in length. This dimension is adapted from the SC structure of OSIRIS-REx, which has an arm of 3.35 m in length.[3]. The simplified polyhedron model of the SC, for the descent phase can be found in Figure 7.5.

Rosetta had bi-propellant (N_2O_4/MMH) thrusters and hence a divided cylindrical fuel tank at the centre of the SC. The upper section of the tank held the propellant whilst the heavier oxidiser was at the bottom section. The tank radius and height are 2.8 m and 1.197 m, respectively. For simplicity we consider the orientation of the observation instruments and actuators to be fixed with respect to the body. For the descent phase, a reaction thruster generating a force in the positive Z-axis of the body reference frame is required and the reaction thrusters will be aligned to fire in the negative Z-axis of the body ref frame.

7.2.2. Mass Properties

The simulator is designed to treat the phases of close observation and descent. The mass properties of Rosetta at these stages is used as a reference. At the start of the mission, Rosetta has a wet mass of 3100 kg and a dry mass of 1380 kg including the solar arrays and the Philae lander. The solar arrays weigh 75 kg each. At the comet rendezvous, approximately 776 m/s of Δv had been used. Initially the two-part cylindrical fuel tank

[3]https://en.wikipedia.org/wiki/OSIRIS-REx, date accessed: 29-02-19

contained 660 kg of propellant of density 880 kg/m^3 and 1060 kg of oxidiser of density 1440 kg/m^3.[4]Using Tsiolkovski's equation the mass of the SC at 100 km altitude insertion can be found. Tsiolkovski's equation is given as below:

$$m_0 = m_t e^{-\frac{\Delta v}{v_e}} \tag{7.3}$$

where m_t is the total initial mass of the SC, m_0 is the remaining total mass of SC after using Δv with a reaction thruster having an exhaust velocity of v_e. For Rosetta v_e was approximately 2200 m/s. Using the masses provided above, the mass of the SC after Δv usage of 776 m/s is $\approx$ 2108 kg. The usual propellant to oxidiser ratio for $CH_4N_2H_3/MON$ is around 1.65 and using this we can find out their remaining masses to be $\approx$ 275 kg and 453 kg, m_p and m_o, respectively. The inertia tensor of the SC can, therefore, be computed in three parts, the block, the solar arrays and the fuel tank. The inertia tensor of the SC is time variable due to the usage of propellant and oxidiser and therefore this change can be implemented by simply changing the inertia tensor by updating the propellant mass and dimensions as per the fuel usage requirements.

Block

$$\boldsymbol{J}_b = \begin{bmatrix} \frac{1}{12}m_b(y_b^2+z_b^2) & 0 & 0 \\ 0 & \frac{1}{12}m_b(x_b^2+z_b^2) & 0 \\ 0 & 0 & \frac{1}{12}m_b(x_b^2+y_b^2) \end{bmatrix} = \begin{bmatrix} 1214 & 0 & 0 \\ 0 & 1256 & 0 \\ 0 & 0 & 862 \end{bmatrix} \text{kg m}^2 \tag{7.4}$$

Solar Arrays

$$\boldsymbol{J}_s = \begin{bmatrix} \frac{1}{12}m_s(y_s^2+z_s^2)+m_s x_d^2 & 0 & 0 \\ 0 & \frac{1}{12}m_s(x_s^2+z_s^2) & 0 \\ 0 & 0 & \frac{1}{12}m_s(x_s^2+y_s^2)+m_s z_d^2 \end{bmatrix} = \begin{bmatrix} 7333 & 0 & 0 \\ 0 & 33 & 0 \\ 0 & 0 & 7300 \end{bmatrix} \text{kg m}^2 \tag{7.5}$$

$$\tag{7.6}$$

Bi-propellant Tank

$$\boldsymbol{J}_p = \begin{bmatrix} \frac{1}{12}m_p(h_p^2+3r^2)+m_p(\frac{h_p}{2})^2 & 0 & 0 \\ 0 & \frac{1}{12}m_p(h_p^2+3r^2)+m_p(\frac{h_p}{2})^2 & 0 \\ 0 & 0 & \frac{1}{2}m_p(h_p^2+3r^2) \end{bmatrix} \text{kg m}^2$$
$$\boldsymbol{J}_o = \begin{bmatrix} \frac{1}{12}m_o(h_o^2+3r^2)+m_o(h_t-\frac{h_o}{2})^2 & 0 & 0 \\ 0 & \frac{1}{12}m_o(h_o^2+3r^2)+m_o(h_t-\frac{h_o}{2})^2 & 0 \\ 0 & 0 & \frac{1}{2}m_o(h_o^2+3r^2) \end{bmatrix} \text{kg m}^2 \tag{7.7}$$

$$\boldsymbol{J}_t = \boldsymbol{J}_p + \boldsymbol{J}_o = \begin{bmatrix} 795 & 0 & 0 \\ 0 & 795 & 0 \\ 0 & 0 & 131 \end{bmatrix} \text{kg m}^2 \tag{7.8}$$

The inertia tensor is a sum of all the above and it comes to:

$$\boldsymbol{J}_{total} = \boldsymbol{J}_b + 2\boldsymbol{J}_s + \boldsymbol{J}_t = \begin{bmatrix} 16675 & 0 & 0 \\ 0 & 2117 & 0 \\ 0 & 0 & 15593 \end{bmatrix} \text{kg m}^2 \tag{7.9}$$

$$\tag{7.10}$$

where, $m_b = 1380 - 150 = 1130$ kg and $m_s = 150$ kg. The inertia computed in Razgus (2017) for the same spacecraft, without the consideration of a cylindrical bi-propellant tank, with 2100 kg remnant mass after orbit insertion at 100 km and a v_e of 2500 m/s, is $J_{11} = 16590$, $J_{22} = 2057$ and $J_{33} = 15964$ kg m^2. The percentage differences of this inertia from the above implemented model, in the X, Y and Z-axes are, 0.5%, 2.92% and 2.31%, respectively. This is logical, since the maximum change with the consideration of bi-propellant tank is along the Z-axis. The percentage along the Y-axis is high due to the lower inertia along the axis. The actual value of inertia change along the X and Y-axis are similar. A better model than the one implemented would consider inertia of the SC block, with a cylindrical empty space for the bi-propellant tank.

[4]http://www.astronautix.com/n/n2o4mmh.html, date accessed:12-04-2018

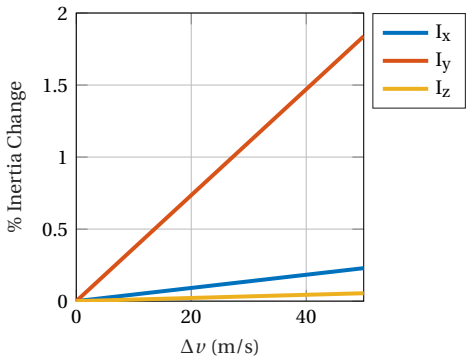


Figure 7.6: Percentage inertia change upto Δv 50 m/s.

The Δv requirements for proximity operations of small Solar System bodies is in the range of cm/s to m/s, depending on their sizes. The biggest asteroid considered in this thesis is Kleopatra. With basic orbital velocity calculations, it can be approximated to have a requirement of 10-50 m/s for the mapping phase, where the orbits range from 10-20 km in altitude. We compute the inertia of the SC (inertia update method explained below), with a Δv usage of 50 m/s and with folded and steered solar panels,

$$\boldsymbol{J}_{s_2} = \begin{bmatrix} 916 & 0 & 0 \\ 0 & 33 & 0 \\ 0 & 0 & 949 \end{bmatrix} \text{kg m}^2, \qquad \boldsymbol{J}_{t_2} = \begin{bmatrix} 752 & 0 & 0 \\ 0 & 752 & 0 \\ 0 & 0 & 122 \end{bmatrix} \text{kg m}^2 \tag{7.11}$$

$$\boldsymbol{J}_{total_2} = \boldsymbol{J}_b + 2\boldsymbol{J}_{s_2} + \boldsymbol{J}_{t_2} = \begin{bmatrix} 3798 & 0 & 0 \\ 0 & 2073 & 0 \\ 0 & 0 & 2881 \end{bmatrix} \text{kg m}^2 \tag{7.12}$$

The objective of the mapping phase is to provide with a required Δv from optimal mapping trajectories and this provides the initial inertia of the SC for the descent phase.

Inertia Update with Time

The inertia tensor is not constant as discussed above and changes with each manoeuvre for mapping as well as during descent. An initial analysis of the inertia tensor is done for varying Δv s to understand if this update is a requirement or can be assumed to be constant at the computed initial inertia for the time-frame of each mapping and descent phases.

A simple model has been used for this analysis. As the propellant or oxidiser is consumed, its level or height in the tank reduces whilst occupying the whole diameter of the tank. The assumption is, therefore, to change the dimension of the tank by simply reducing its height. The treatment of fuel sloshing has already been ruled out of scope for this thesis and therefore, the tank size is always considered to be snug to the volume of available propellant or oxidiser at any instant.

Using Tsiolkovski's equation as given in Eq. (7.3) the mass can be updated with Δv usage. This can then be used to determine the new inertia of the SC. Due to the weak gravitational attraction of asteroids, the Δv usage for observation and descent phases is within the range of a 50 m/s (for Kleopatra, less than 10 cm/s for Itokawa and Comet 67P). We compute percentage change in inertia due to a range of Δv as shown in Figure 7.6 to decide if it is required to have inertia update every time step or if we can use constant inertia for each phase of the mission.

Coriolis Force and Moment due to Variable Mass and Rotational Rates

The variability in SC mass, while the engine expels propellant and rotates at a finite speed exerts a force on the SC. From Table 2.2, the maximum mass flow rate can be found to be ≈0.084 kg/s for a 240 N thrust. The maximum rotational speed is constrained to 20°/s or 0.35 rad/s and the distance from the com is 1.5 m. From this the Coriolis force can be computed using (Mooij (1994)),

$$\boldsymbol{F}_C = -2\boldsymbol{\omega} \times (\dot{m}\boldsymbol{r}_e) \tag{7.13}$$

The maximum force can be calculated as, $-2 \times 0.35 \times 0.084 \times 1.5 = 0.0882$ N. This force is very small as compared to the minimum allowable thrust and therefore, can be ignored. The Coriolis moment can be computed

as (Mooij (1994)),

$$\boldsymbol{M}_C = -\frac{\delta \boldsymbol{J}}{\delta t} \cdot \boldsymbol{\omega} - \dot{m} \boldsymbol{r}_e \times (\boldsymbol{\omega} \times \boldsymbol{r}_e) \tag{7.14}$$

As can be seen, the % change in inertia for the entire mapping phase, which lasts for two days is ≈0.8% on average for the three axes. The rate of change of inertia with time is therefore negligible. So, the first term in the RHS can be ignored. The second term in the RHS can be computed to be a maximum of 0.066 Nm, which is also very small. Hence, effects on inertia due to variable mass and rotational rates can be ignored.

7.2.3. Reflectivity

The reflectivity of the SC is a key factor to model the SRP acting on the SC as can be seen from Eq. (4.118) in Chapter 4. The SC is divided into two structures: the main body with twelve triangular facets and the solar arrays with ten triangular facets each on either side of the main body as can be seen in Fig 7.4. The solar panels are considered to have zero thickness. The SC, therefore, has a total of 32 facets, which is representative enough to model the SRP. We compute the normals from each of these facets and their centroids to have their distances from the centre of mass of the SC in the body reference frame, `B` at all times.

We consider only the specular reflectivity of the SC and not its diffuse reflectivity. Specular reflection is nothing but the regular reflection where each incident ray is reflected at the same angle to the surface normal as the incident ray, but on the opposite side of the surface normal on the plane formed by the incident and reflected rays.[5] and diffuse reflection is the reflection of light or other waves or particles from a surface such that a ray incident on the surface is scattered at many angles rather than at just one angle as in the case of specular reflection.[6]. We use the values given in Montebruck and Gill (2008) for the reflectivity of solar panels, which is $\alpha_{r_p} = 0.21$ and consider the body to be more reflective with a value of $\alpha_{r_b} = 0.5$.

7.3. Asteroid Model

The asteroid model holds information about the physical properties of the asteroid. As per our assumptions in Chapter 2, the SC should have a priori knowledge about the shape to a certain resolution, mass, angular velocity, mean density, gravity and a landmark map of the asteroid of interest before the close proximity operations (mass, angular velocity, mean density and gravity are updated to a better accuracy with the mission phases before close proximity operations). This is sufficient for the guidance algorithm to generate safe trajectories for the close proximity operations. With the mapping phase the resolution of the asteroid shape is enhanced. It also helps in selecting a landing site for the descent phase. The small Solar System bodies used for the thesis will be Kleopatra and Itokawa. Their properties have been listed in Table 2.1.

7.3.1. Polyhedron Models

A polyhedron model, as the name suggests, captures the shape of the body of interest by means of polyhedral shapes. In the case of the surface structure of irregularly shaped celestial bodies, these models consist of triangular facets meshed together. As can be intuitively guessed the smaller the size of these facets, a better resolution of the shape is obtained. But this also means a higher number of facets and this increases computational load as well as more memory storage capacity. The resolution also depends on whether the data is generated by ranging measurements from Earth or from flyby or close proximity missions to them.

The polyhedron.txt files for the selected asteroids for this thesis are available on the Jet Propulsion Laboratory website.[7] These files contain the vertices of the facets. The facets are indexed in a counter-clockwise fashion when observed from outside, making the computations of normals from them logical. Figure 7.7 gives the polyhedral surface structures of Kleopatra, Itokawa and Comet 67P, respectively. Higher resolution polyhedral models are also available as shown for Comet 67P, but they are available for asteroids, that have already been visited in a mission. For the not visited small Solar System bodies, Earth ranging measurements are used to build a model, resulting in a lower resolution.

For comparison, Itokawa polyhedron files with three resolutions have been run. The CPU time for extracting the data from the .txt file and building the polyhedron model has been provided in Table 7.1, Figure 7.8, is a plot of the percentage accuracy of computed gravitational acceleration versus the computation cost for 50 SC positions. 50 SC positions are used, since this is a generic number of nodes used for descent phase. The CPU time is multiplied by 10, since this is the maximum number of iterations for convergence of the

[5] https://en.wikipedia.org/wiki/Specular_reflection, date accessed:02-05-18
[6] https://en.wikipedia.org/wiki/Diffuse_reflection, date accessed:02-05-18
[7] https://echo.jpl.nasa.gov/, date accessed: 14-02-18

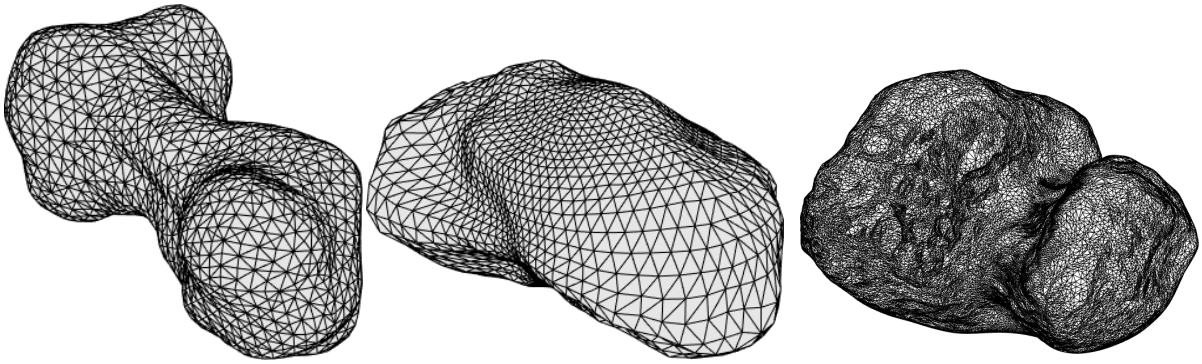


Figure 7.7: Polyhedron models of Kleopatra (left), Itokawa (middle) and Comet 67P (left) (not to scale). As can be seen, the actual structure of Comet 67P has be replicated with a higher mesh of polyhedrons.

Table 7.1: Itokawa polyhedron files and their CPU time for computation

File Nr.	Vertices	Facets	Edges	CPU time
1	1846	3688	5532	0.3234
2	25350	49152	73728	27.92
3	99846	196608	249412	1248.53

SCvx algorithm. The third file, with maximum number of polyhedrons is considered as 100% accurate while computing the percentage accuracy of the gravitational acceleration. As can be seen there is a substantial increase in the CPU time, when the number of polyhedrons are increased by a factor of 10 for each file. Since, the gravitational acceleration is still of the same order for all three files, the lowest resolution files available for the asteroids are chosen. Also, the time taken for the SCvx algorithm to run all its iteration is in 10s of secs, which is an added factor to keep the polyhedral computation time minimal. Note that this computation time can be reduced by using .mex files, which has been done to compute the polyhedron gravity field for Kleopatra in Section 7.4.4.

7.3.2. Landmark Map

The purpose of the mission phase of mapping the asteroid surface as discussed in Appendix A is to generate a topographic map of the asteroid surface to help select sites of interest for the descent phase. These sites are selected based on surface features like craters, ridges, rocks or other visually recognisable features. The sampling based motion planning guidance is to find the optimum trajectories to observe the entire asteroid

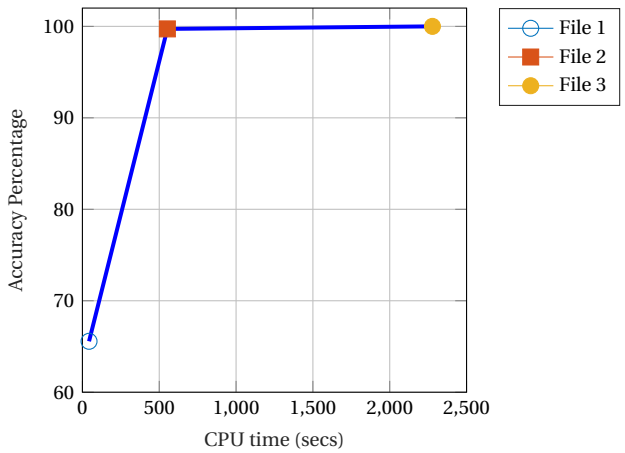


Figure 7.8: Accuracy of computed polyhedron gravity compared to CPU computational time for 50 nodes and 10 iterations.

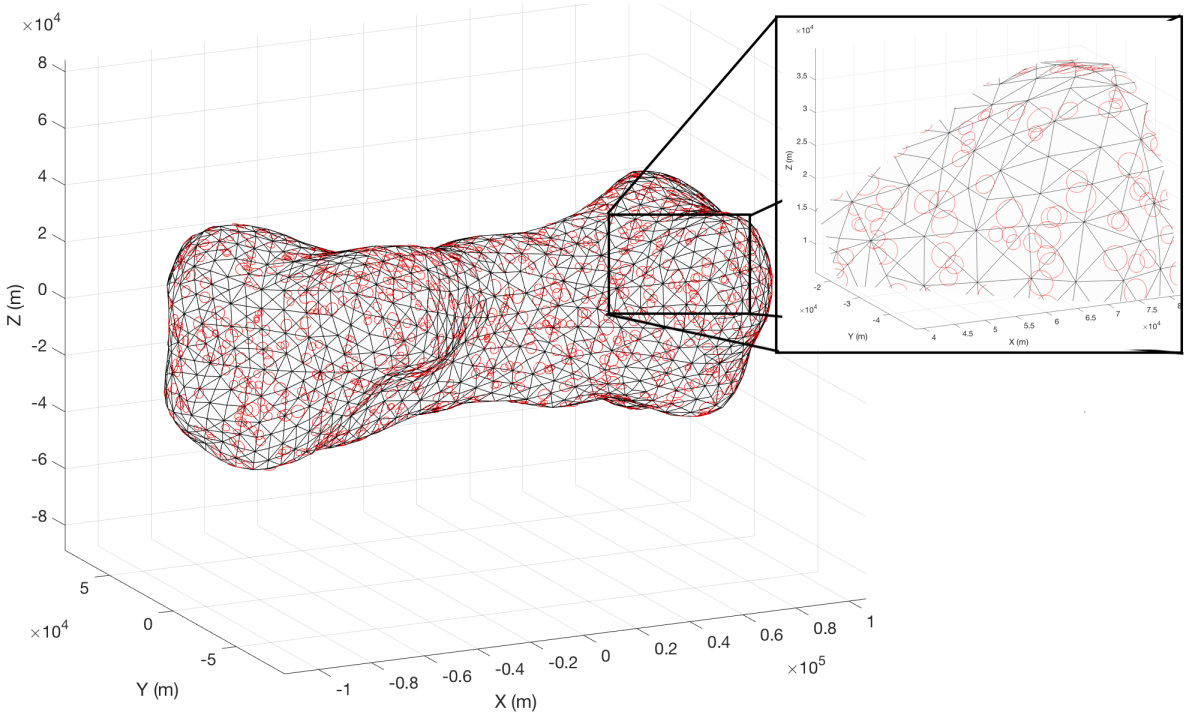


Figure 7.9: Pre-generated landmark map of Kleopatra (2000 landmarks).

surface while finding a golden mean between the consumption of fuel and mission time. Since no actual SC trajectory manoeuvre takes place during the autonomous guidance searches for this trajectory we do not need the visual-based navigation system to frequently measure the relative position of the SC. The SC just requires a generic landmark map to realise the surface of the asteroid it covers to analyse the computed trajectories and help it search for the optimal trajectory.

Taking real images and doing image processing once the trajectory is selected is not in the scope of this thesis. This is another reason for the requirement of an a priori landmark map. For the guidance subsystem, we simply require the position of such landmarks with respect to the asteroid frame. Assuming perfect knowledge of the position and attitude of the SC with the help of the navigation subsystem, its ground-track on the surface of the asteroid can be generated.

Generating landmarks on the polyhedral model is simple, since polyhedrons are convex by nature and can be treated as individual convex hulls. Since surface features on an asteroid are not evenly spread, the landmarks have to be randomly placed on each polyhedron of the asteroid surface. This can be done by the formula (Osada et al. (2002)):

$$\boldsymbol{r}_{LM}^{\mathtt{A}} = (1-\sqrt{p_1})\boldsymbol{A} + \sqrt{p_1}(1-\sqrt{p_2})\boldsymbol{B} + \sqrt{p_1 p_2}\boldsymbol{C} \tag{7.15}$$

where, $\boldsymbol{r}_{LM}^{\mathtt{A}}$ is the position vector of the landmark with respect to the $\mathtt{A}$-frame, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are the vertices of a triangular polyhedron on the asteroid surface and p_1, p_2 are the percentages from the vertex A to the opposite edge and along the edge, respectively. By random selection of p_1 and p_2 between [0,1], random locations from the vertex $\boldsymbol{A}$ can be selected thereby providing us with random landmarks.

Pre-generated landmark maps that have been used for different research works use different densities of landmarks per km^2. Razgus (2017), has reviewed these works and found the number of landmarks to lie between 100 to 5000 for certain reviewed asteroids. For this thesis, 2000 landmarks are generated as an example and can be seen in Figure 7.9. It can be also be seen from the figure, that they are randomly distributed with varied density over the polyhedrons. The number of landmarks can be chosen by the user, but the density is predefined by a ratio to put more landmarks or larger polyhedrons.

7.3.3. Illumination Model

Mapping phase of the mission requires information about the surface illumination of the asteroid. This enables the guidance algorithm to make intelligent choices of the trajectories. Since the illumination of the

surface is dependent on the position of the Sun with respect to the target body, its rotation rate and albedo, this model is time variant.

Sun Position
The angle subtended by the Sun on asteroids of sizes in the range of 100 to 1000 km in MAB (2.1 to 3.3 au) can be calculated roughly to be between $0.01^\circ \times 10^{-3}$ to $0.2^\circ \times 10^{-3}$. These angles are too small and would not change the illumination of the asteroid by much as long as the orbiting motion of the target body about the Sun is limited within a few kilometres. The Sun position can, therefore, be fixed since the asteroids move less than a degree per day and the illumination intensity due to the Sun position can be kept constant.

Albedo
Albedo is the amount of diffuse reflection from the surface compared to the total solar irradiation of the surface. It is a surface property and is not extensively dependent on the amount of illumination of the surface, since if the solar irradiation is less the diffuse reflection would reduce by proportion for the whole surface. The albedo would be higher in the presence of icy surfaces, but this cannot be specifically modelled and neither its dependence on the phase angle. As discussed in Section 5.2.1, the Lambertian model is inaccurate, but should suffice for the guidance algorithm. This is because the guidance algorithm is independent of the accuracy of the illumination model and should give trajectories based on the most illuminated areas to the least illuminated areas. Addition of albedo to the illumination model will therefore be left for future work.

Asteroid Rotation and Inclination
As the asteroid rotates, the angle made by solar irradiation and the surface changes and hence the illumination. The asteroid inclination also effects the illumination and since we assume the asteroid to not be tumbling, the surface normals are already in the A-frame and therefore independent of inclination.

The illumination model is, therefore, time dependent due to the asteroid rotation rate only. The normals from the facets in the polyhedral model of the asteroid are already calculated in the asteroid model and the Sun position is fixed. The illumination map is created by the angle between the surface normal and the Sun position in the I-frame. For the guidance algorithm this map converted to the relative frame. The change in the angle between the surface normals and the Sun position can be found by the change in $\check{\boldsymbol{q}}_{\mathtt{A/I}}$. The time-dependent angle between the normals and the Sun position is calculated as below,

$$\cos\theta_{vis} = \frac{n_f^{\mathtt{I}}.\boldsymbol{r}_{sun}^{\mathtt{I}}}{||\boldsymbol{r}_{sun}^{\mathtt{I}}||} \tag{7.16}$$

The value of this ranges from $-1 \leq \cos\theta_{vis} \leq 1$, which is perfect to generate the visibility map. The normals that make greater than 90° with the Sun direction are converted to zero and their respective polyhedron are the darkest and as the angle reduces, we get lighter regions thus depicting visibility of the same. Figure 7.10 shows the illumination map at the initial state (t_0) at orbit insertion of the SC. This comprises the illumination model providing the guidance algorithm with the information to make intelligent decisions during mapping.

7.4. Dynamics Model

The dynamics of the system are a part of the simulations block as discussed in Section 7.1.2. The EOMs in the relative frame using DQs have been derived in Chapter 4. Further they are linearised in Chapter 6. These equations need to be verified. The external disturbance models discussed in Chapter 4 also need to be verified and these are done by the DQ-integrator verified in the the first subsection. The polyhedral gravity model is used as is from Razgus (2017), but it is still verified to makes sure its functionality is correct.

7.4.1. Translational and Rotational Motion

To verify the derived EOMs, an integrator with Runge-Kutta 45 (RK45) has been used.[8] The integrator contains the EOMs of 6DOF motion with Cartesian coordinates and quaternions in the inertial frame, I, (Q-Integrator), which is converted to relative frame, A, using quaternion rotation and those with DQs in the relative frame, A, (DQ-Integrator). For the purpose of verification we consider an example with a central point-mass gravity field and in the absence of perturbing forces and torques. This would lead to a normal and precise Keplerian orbit with constant energy.

[8] Trade off for available MATLAB® integrators can be found in Appendix C

Test Case: Sphere of radius (target body): 50 km, Mean density: 5000 kg/m^3, Semi-major axis, a: 60 km, Eccentricity: 0, Inclination: 0°. The mass of the target body is, therefore, 2.618×10^{18} kg. Using the gravitational constant, G as 6.67×10^{11} kg m^2. We consider the A frame to be rotating about the Z-axis of the I frame with a constant angular velocity of 3×10^{-4} rad/s and the B frame rotating about its X-axis with 6.28318×10^{-4} rad/s.[9] The gravitational acceleration, the circular velocity at the altitude of 10 km (both in the asteroid centre fixed inertial frame) and the period of the orbit can be given by the formulae:

$$\boldsymbol{g}_{\mathtt{I}} = G\frac{M_{\text{target}}}{r^3}\boldsymbol{r}_{\mathtt{I}} = 0.048535 \text{ m/s}^2, \quad \boldsymbol{v}_{\text{circ}} = \sqrt{\frac{GM_{\text{target}}}{a}} = 53.964 \text{ m/s}, \quad T = 2\pi\sqrt{\frac{a^3}{GM_{\text{target}}}} \approx 6986 \text{ s} \tag{7.17}$$

We simulate the equatorial orbit for 20,000 secs with the initial conditions as given in Table 7.2. Figure 7.11 presents the results obtained from the simulation for the inertial frame EOMs. We do this to make sure the Q-Integrator is running fine by comparing the results from Razgus (2017). The plots contain the position (I-frame) of the SC orbiting the asteroid in a point mass gravity field, the angular velocity of the B frame with respect to I frame and the quaternions representing the orientations of the B and A frames with respect to the I frame.

Table 7.2: Problem Parameters for Dynamics Verification

$\boldsymbol{r}_0^{\mathtt{I}}$ (km)	$\boldsymbol{v}_0^{\mathtt{I}}$ (m/s)	$\boldsymbol{\omega}_{\mathtt{A/I}}^{\mathtt{I}}$ (rad/s)	$\boldsymbol{\omega}_{\mathtt{B/I}}^{\mathtt{B}}$ (rad/s)	$\boldsymbol{q}_{\mathtt{B/I}_0}$	$\boldsymbol{q}_{\mathtt{A/I}_0}$
$[0,60,0]^{\mathtt{T}}$	$[53.964,0,0]^{\mathtt{T}}$	$[0,0,3\times10^{-4}]^{\mathtt{T}}$	$[6.28\times10^{-4},0,0]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$

Table 7.3: Problem Parameters for DQ Integrator Verification

$\check{\boldsymbol{q}}_{\mathtt{B/A}_0}$ (-, km)	$\check{\boldsymbol{\omega}}_{\mathtt{B/A}_0}^{\mathtt{B}}$ (rad/s, m/s)	$\check{\boldsymbol{\omega}}_{\mathtt{A/I}}^{\mathtt{I}}$ (rad/s, m/s)
$[0,0,0,1,0,30,0,0]^{\mathtt{T}}$	$[6.28\times10^{-4},0,3\times10^{-4},0,0,73.411,0,0]^{\mathtt{T}}$	$[0,0,3\times10^{-4},0,0,0,0,0]^{\mathtt{T}}$

Since the initial conditions suggest an orbit in the equatorial plane (X-Y plane), the Z-axis component of the SC in the I frame remains zero throughout the duration of approximately three orbital periods. Also with respect to the inertial frame, the angular velocity of the B frame remains constant. The value of total energy remains constant at -1455.172 J, with an accuracy of 7.7×10^{-6}. Hence the Q-Integrator is verified and can be used to generate the orbits and angular velocities in the A frame and B frame. These are then used to verify the DQ-integrator in the A frame and B frame.

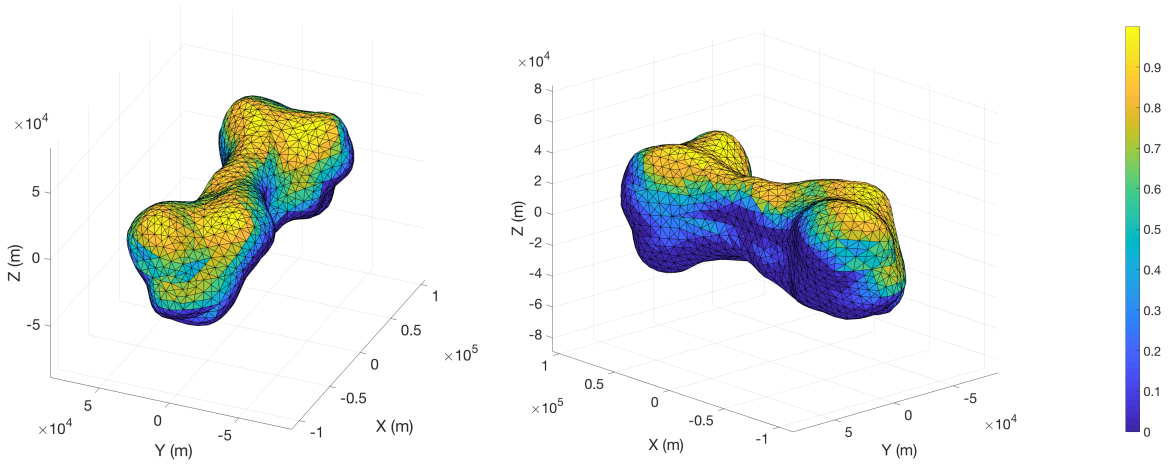


Figure 7.10: Illumination map of the asteroid Kleopatra with cold to warm gradient for minimum to maximum illumination.

[9] The test case is the same as Razgus (2017) so that a one to one verification can be done

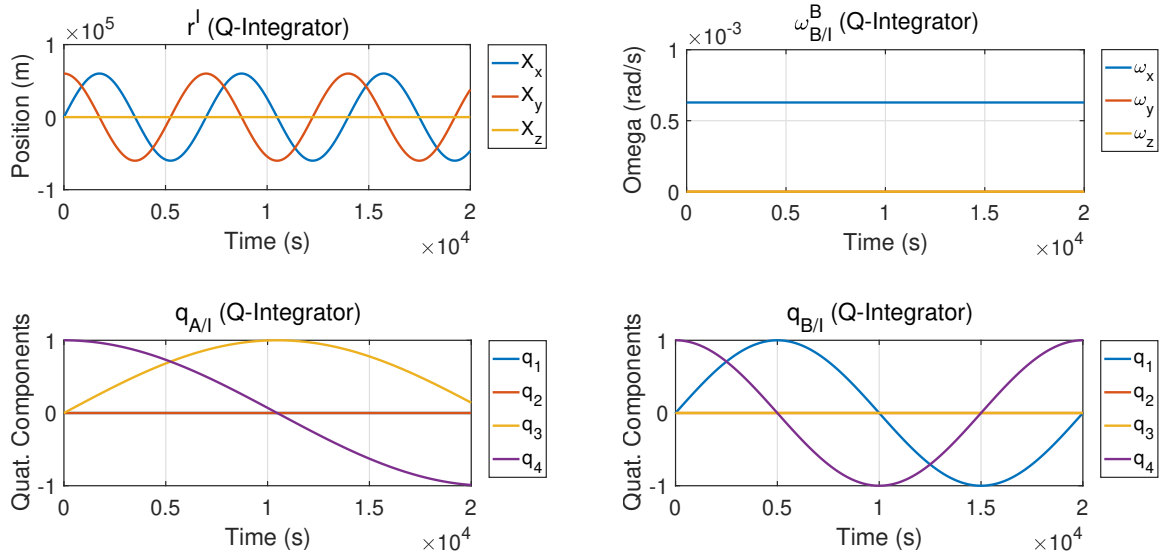


Figure 7.11: Simulated position, angular velocity and quaternions with respect to the I frame.

Figure 7.12 to 7.16 plots the results from both the Q-integrator converted by DCMs to the A-frame and DQ-integrator converted by DQ arithmetic to Cartesian coordinates and quaternions. The plots include the position of the SC in the A and B frames, the translational velocities in the B frame, the angular velocities in the B frame and the quaternion representing the orientation of the B frame to the A frame. As can be seen from the plots, the results of all the variables from the DQ-integrator follow the same patterns as those from the Q-integrator. Figure 7.17, is plotted to give an idea of the accuracy of the DQ plots. The SC position in the A and B frame for both integrators have been overlaid to confirm the pattern and a magnification of them is provided to provide the accuracy achieved with the DQ-integrator.

From the overlay of the SC position, no visible difference in the plots can be seen and hence, we also plot the magnification. As can be seen from the magnification the difference between the results from the integrators is within the range of 4×10^{-5} m, for both position representations, which is well within the acceptable error for this thesis. This verifies the DQ-integrator and in turn verifies the derived DQ dynamics and kinematics derived in Section 4.9.2 and Section 4.9.3.

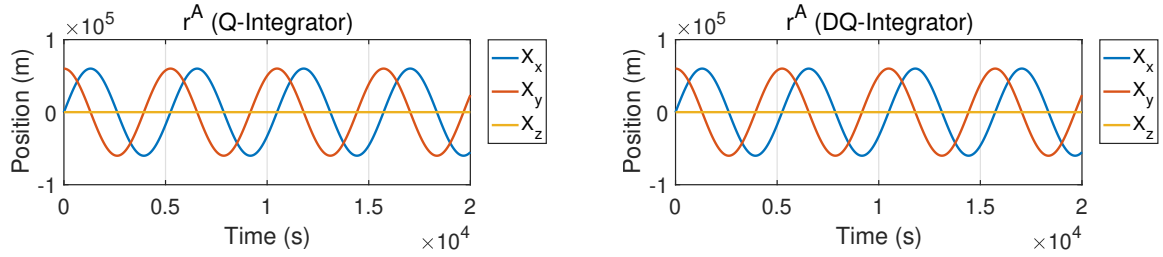


Figure 7.12: Simulated motion in the A frame from Inertial integrator (left) and DQ integrator (right).

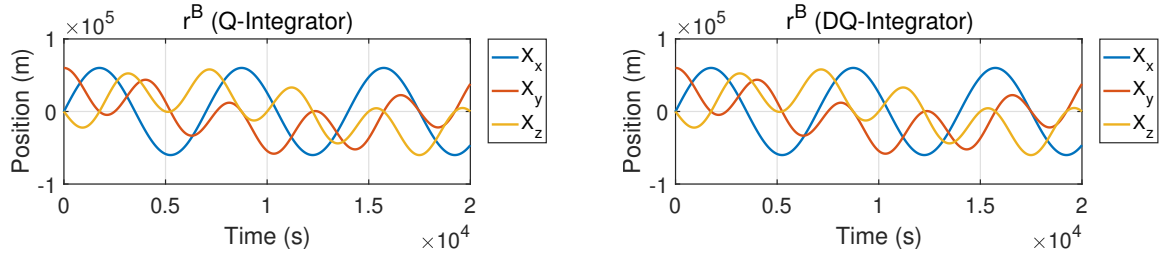


Figure 7.13: Simulated motion in the B frame from Inertial integrator (left) and DQ integrator (right).

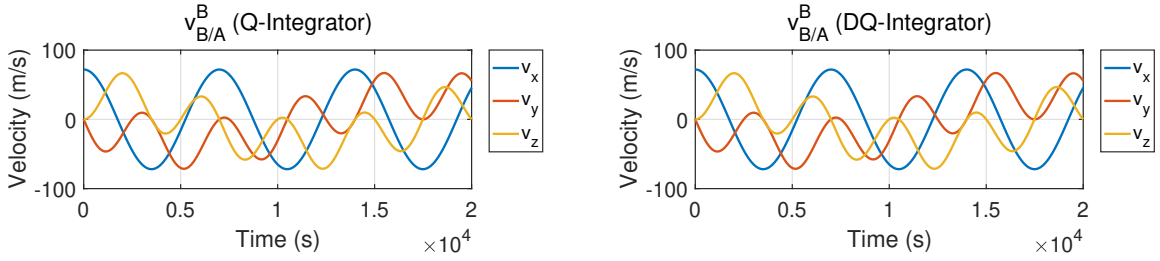


Figure 7.14: Simulated relative velocity, $\boldsymbol{v}^{\mathtt{B}}_{\mathtt{B/A}}$ from Inertial integrator (left) DQ and integrator (right).

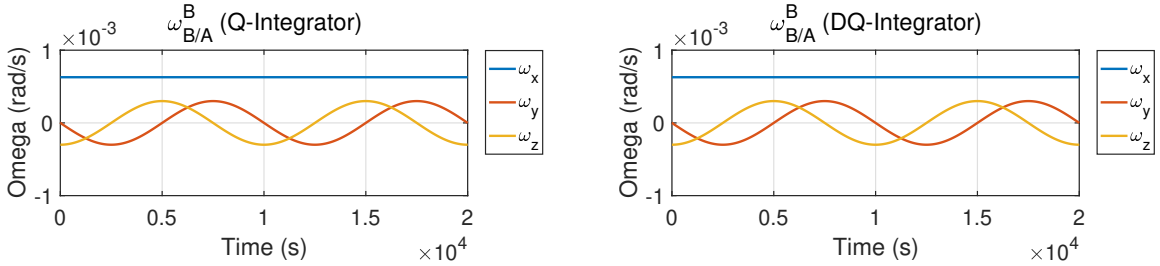


Figure 7.15: Simulated angular velocity, $\boldsymbol{\omega}^{\mathtt{A}}_{\mathtt{B/A}}$ from Inertial integrator (left) and DQ integrator (right).

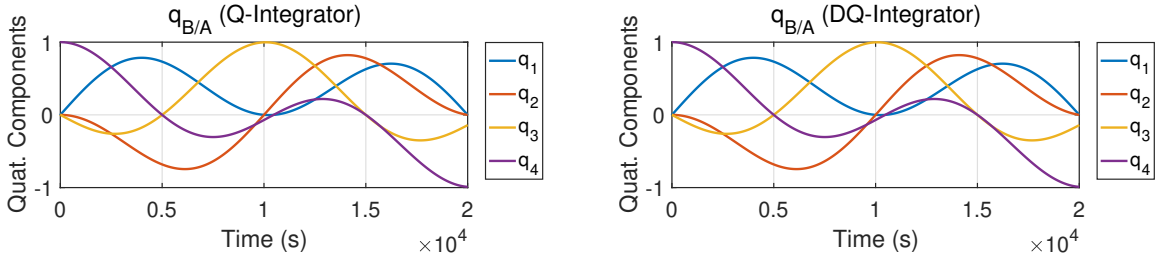


Figure 7.16: Simulated angular velocity, $\boldsymbol{q}_{\mathtt{B/A}}$ from Inertial integrator(left) DQ integrator(right).

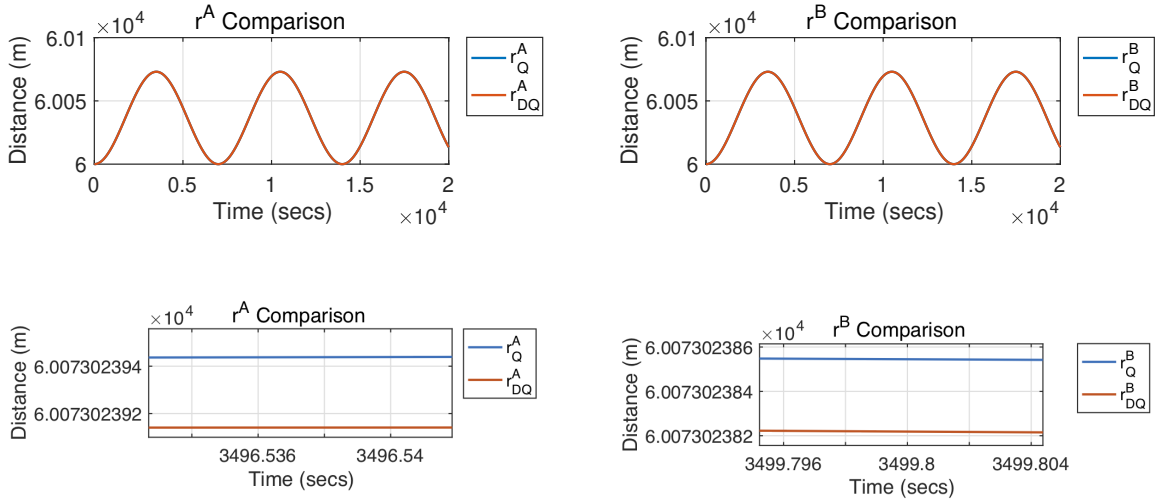


Figure 7.17: Overlay of $\boldsymbol{r}^{\mathtt{A}}$ and $\boldsymbol{r}^{\mathtt{B}}$ from the Q and DQ integrator and the magnified images to view the error. The difference is within the range of 4×10^{-5} m

7.4.2. Solar Radiation Pressure

As shown in Figure 7.4, the SC is made of N plates/faces having an area of A_i and reflectivity α_i. The analytical equation given in Section 4.11.2 gives the sum of the SRP on individual plates. To verify that the mathematical model of the SRP works correctly we use the variation equations given in Wakker (2015). The equations for maximum variation in the orbital elements are as below (Wakker (2015)),

$$|\Delta a|_{max} = 4\frac{f}{n_0^2}\cos\alpha_0 \tag{7.18}$$

$$|\Delta i|_{max} = \frac{f}{n_0^2 r_0}\sin\alpha_0 \tag{7.19}$$

$$|\Delta\Omega|_{max} = 2\frac{f}{n_0^2 r_0}\sin\alpha_0 \tag{7.20}$$

where, a, i & Ω are the semi-major axis, inclination and right ascension of the ascending node of a Keplerian orbit. f is the magnitude of acceleration caused by SRP, r_0 is the orbital radius, n_0 the mean motion in orbit and α_0 the angle between the orbital plane and Sun direction. We use the same target body as used in Section 7.4.1, but at the distance of the asteroid Kleopatra, 2.7941 au. The solar flux at the distance of the Earth is 1361W/m^2 and can be found at this distance by the inverse square law. The area of the SC as discussed in Section 7.2.1 is $\approx$ 95m^2 and its mass 2100 kg. A polar orbit around the target body is considered and the SC attitude is fixed to keep the solar panels facing the Sun during the period of simulation.

Table 7.4: Problem Parameters for SRP Verification

$\boldsymbol{r}_0^{\mathtt{I}}$ (km)	$\boldsymbol{v}_0^{\mathtt{I}}$ (m/s)	$\boldsymbol{\omega}_{\mathtt{A/I}}^{\mathtt{I}}$ (rad/s)	$\boldsymbol{\omega}_{\mathtt{B/I}}^{\mathtt{B}}$ (rad/s)	$\boldsymbol{q}_{\mathtt{B/I}}$	$\boldsymbol{q}_{\mathtt{A/I}}$
$[0,0,100]^{\mathtt{T}}$	$[0,-41.79,0]^{\mathtt{T}}$	$[0,0,3\times10^{-4}]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$

For this configuration, numerically the acceleration, $f \approx 2.680\times10^{-8}$m/s^2, the mean motion, $n_0 = 4.179\times10^{-4}$ rad/s and $\alpha_0 = 90°$. The maximum variations for this orbit are,

$$|\Delta a|_{max} = 0 \text{ m} \tag{7.21}$$

$$|\Delta i|_{max} = 8.794° \times 10^{-5} \tag{7.22}$$

$$|\Delta\Omega|_{max} = 1.759° \times 10^{-4} \tag{7.23}$$

The results from the simulation give the values of 2×10^{-7}, 8.198×10^{-5} and 1.639×10^{-4} for $|\Delta a|_{max}$, $|\Delta i|_{max}$ and $|\Delta\Omega|_{max}$, respectively. Hence, we can conclude that the SRP model has been correctly implemented. Figure 7.18 shows the variation in semi-major axis, inclination and RAAN over the period of simulation for the problem configuration.

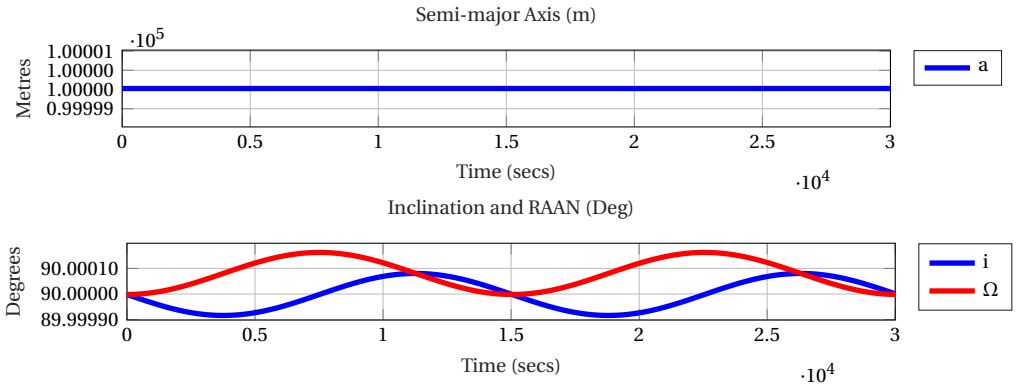


Figure 7.18: Orbital element variations due to SRP force (simulations). The variations are 2×10^{-7}, 8.198×10^{-5} and 1.639×10^{-4} for $|\Delta a|_{max}$, $|\Delta i|_{max}$ and $|\Delta\Omega|_{max}$, respectivetly.

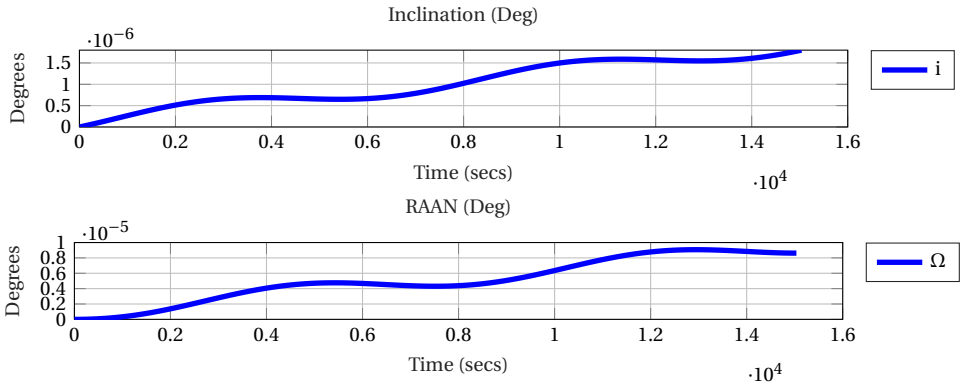


Figure 7.19: Orbital element variations due to 3rdBP force (simulations). The variations are 1.799×10^{-06} and 9.067×10^{-06} for $|\Delta_{2\pi} i|_{max}$ and $|\Delta_{2\pi}\Omega|_{max}$, respectively.

7.4.3. Third Body Perturbation

The implementation of third-body perturbation is as discussed in Section 4.11.3. The acceleration due to this is in the `I` frame and we need to convert it to the `B` frame by means of the quaternion $\boldsymbol{q}_{\mathtt{B/A}} \otimes \boldsymbol{q}_{\mathtt{A/I}}$. The maximum variation in the inclination and RAAN after a complete orbit (true anomaly = 0 to 2π) of the SC around the target body is given by equations (Wakker (2015)):

$$|\Delta_{2\pi} i|_{max} = \frac{-3}{2}\pi \frac{\mu_d}{\mu}\left(\frac{r_0}{r_d}\right)^3 \sin i_0 \sin 2\alpha_0 \tag{7.24}$$

$$|\Delta_{2\pi}\Omega|_{max} = -3\pi \frac{\mu_d}{\mu}\left(\frac{r_0}{r_d}\right)^3 \cos i_0 \sin^2 \alpha_0 \tag{7.25}$$

where, μ_d, & μ are the standard gravitational parameters for the disturbing body (in our case the Sun) and the target body, respectively and r_d is the distance of the disturbing body from the target body. We know, $\mu_{sun} = 1.3271 \times 10^{20}$ and the distance of the Sun ($r_d = r_{sun}$) from the target body is at a distance of 2.546 au. In order to get values for variations due to an inclined orbit, we consider the orbit to be inclined at $i_0 = 10°$. Then we vary α_0 from 0 to π, to find the maximum value of variation.

Table 7.5: Problem Parameters for 3rdBP Verification

$\boldsymbol{r}_0^{\mathtt{I}}$ (km)	$\boldsymbol{v}_0^{\mathtt{I}}$ (m/s)	$\boldsymbol{\omega}_{\mathtt{A/I}}^{\mathtt{I}}$ (rad/s)	$\boldsymbol{\omega}_{\mathtt{B/I}}^{\mathtt{B}}$ (rad/s)	$\boldsymbol{q}_{\mathtt{B/I}}$	$\boldsymbol{q}_{\mathtt{A/I}}$
$[0,100,0]^{\mathtt{T}}$	$[-41.15,0,7.26]^{\mathtt{T}}$	$[0,0,3\times10^{-4}]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$

For this configuration, the maximum variations are numerically calculated to be,

$$|\Delta_{2\pi} i|_{max} = 6.367° \times 10^{-7} \tag{7.26}$$

$$|\Delta_{2\pi}\Omega|_{max} = 6.713° \times 10^{-6} \tag{7.27}$$

The results from the model give the values of 1.799×10^{-06} and 9.067×10^{-06} for $|\Delta_{2\pi} i|_{max}$ and $|\Delta_{2\pi}\Omega|_{max}$, respectively. As can be seen the ratio of maximum variation of RAAN to that in inclination remains the same. Since the variational equations are approximations, an increase in the results from the model by an order of 10 from the variational equations is accepted. Hence, we can conclude that the 3rdBP model has been correctly implemented. Figure 7.19 shows the variation in the inclination from 10° and RAAN from 90° for a single orbit of the SC around the target body. As can be seen the 3rd body perturbations are an order of 10 less than SRP and an order of 10^{-3} less than the gravitational acceleration as shown in Section 7.4.4. The 3rdBP model can therefore be neglected from the dynamics.

7.4.4. Polyhedron Gravity

The polyhedral gravity model has been provided by Razgus (2017). It has been thoroughly verified and hence will be used as is without any changes. The only change is to convert it to a .mex file to have faster computations of gravity. To make sure that there have been no unintentional changes in the file, the gravity field

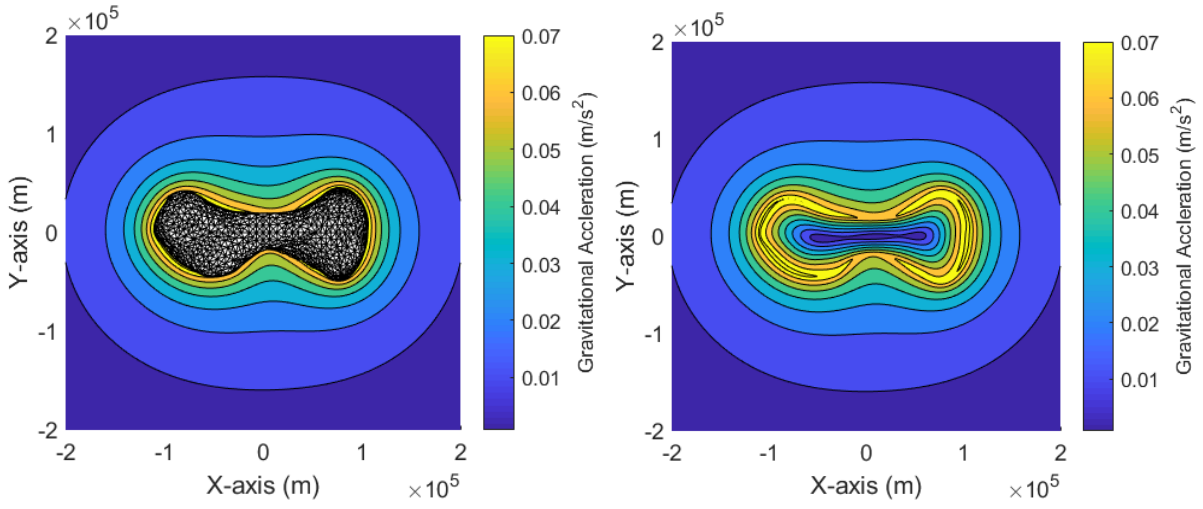


Figure 7.20: Gravity field of Kleopatra in the X-Y plane

of Kleopatra is generated are compared with the results in Razgus (2017). Figure 7.20 shows the plot of the gravity field around Kleopatra to 20000 m altitude, which shows a maximum acceleration higher than Razgus (2017). This is due to the fact, that the standard gravitational parameter for Kleopatra (Table 2.1) is higher than the value used by Razgus (2017).

7.4.5. Gravity Gradient Torque

As discussed in Section 4.11.2, the polyhedron gravity model is implemented to find the gravitational acceleration and using this along with a point mass structure of the SC (point mass at every vertex and centre of each facet) we build the gravity gradient model. To verify this model, we generate the gravity gradient torques for a polar orbit under the gravitational acceleration of Kleopatra for an SC with 80 point masses and compare the plots achieved by Razgus (2017). Figure 7.21 and 7.22 gives a plot gravity gradient torques for the three axes 80 point masses. From the results in Razgus (2017), for gravity gradient torques, it is found that the error percentage between 96 point masses and 175 point masses is ≈ 2%. With 80 point masses, the error percentage would be similar and with a safe margin of 5% error over the obtained gravity gradient torques, the range of errors is 10^{-5} to 10^{-4} Nm. Therefore, 80 point masses are fixed for the purpose of this thesis. The orbit parameters are given in Table 7.6.

Table 7.6: Problem Parameters for Gravity Gradient Verification

$\boldsymbol{r}_0^{\mathrm{I}}$ (km)	$\boldsymbol{v}_0^{\mathrm{I}}$ (m/s)	$\boldsymbol{\omega}_{\mathrm{A/I}}^{\mathrm{I}}$ (rad/s)	$\boldsymbol{\omega}_{\mathrm{B/I}}^{\mathrm{B}}$ (rad/s)	$\boldsymbol{q}_{\mathrm{B/I}}$	$\boldsymbol{q}_{\mathrm{A/I}}$
$[0,0,100]^{\mathrm{T}}$	$[0,-55.64865,0]^{\mathrm{T}}$	$[0,0,3.241\times10^{-4}]^{\mathrm{T}}$	$[4.381,1.854,0]^{\mathrm{T}}\times10^{-4}$	$[0,0,0,1]^{\mathrm{T}}$	$[0,0,0,1]^{\mathrm{T}}$

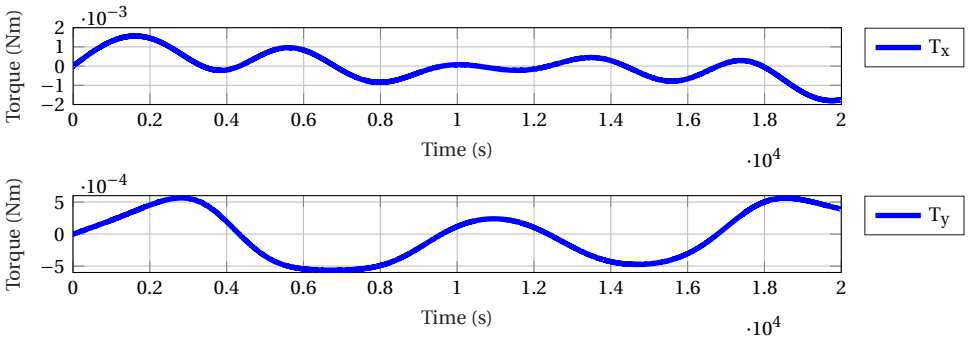


Figure 7.21: Gravity gradient torque, X-axis and Y-axis components at a polar orbit around Kleopatra.

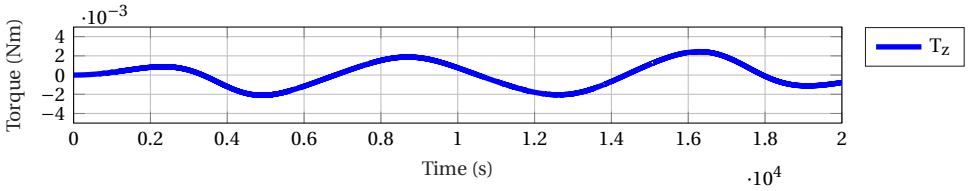


Figure 7.22: Gravity gradient torque, Z-axis component at a polar orbit around Kleopatra

7.5. 6DOF Mars Descent

For proof of concept, we use the Szmuk et al. (2017) test case. First we verify that our linearised dynamics are at par with the actual non linear dynamics of the system. The derivations for the linearised dynamics and the problem formulation for this test case can be found in Appendix B. Next we verify if the SCvx optimiser works as an integrator giving us the same results as the ones obtained from the RK45 integrator for the non-linear and linearised dynamics. Then we verify the results we obtain from our modified SCvx algorithm discussed in Chapter 6 with the results from their research paper. The algorithm and simulation parameters are given in Table 7.7 and 7.8, respectively.

Table 7.7: Algorithm Parameters for Descent Verification

$\boldsymbol{r}_0^{\mathtt{I}}$ (m)	$\boldsymbol{v}_0^{\mathtt{I}}$ (m/s)	$\boldsymbol{r}_f^{\mathtt{I}}$ (m)	$\boldsymbol{v}_f^{\mathtt{I}}$ (m/s)	$\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B}/\mathtt{I}_{0/f}}$ (rad/s)	$\boldsymbol{q}_{\mathtt{B}/\mathtt{I}_{0/f}}$	m_{wet} (kg)	t_f (s)
$[2,1,0]^{\mathtt{T}}$	$[-1,0.2,0]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[-0.1,0,0]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$	2	5

Table 7.8: Simulation Parameters for Descent Verification

m_{dry} (kg)	J (kg m^2)	$\boldsymbol{g}^I$ (m/s^2)	α (s/m)	$\boldsymbol{r}_{\mathtt{T}}^{\mathtt{B}}$ (m)
0.75	diag([0.5,0.5,0.5])	$[-1,0,0]^{\mathtt{T}}$	0.1	$[-1,0,0]^{\mathtt{T}}$

7.5.1. Linearised Cartesian-Quaternion Dynamics (I-frame)

The verification for the linearised dynamics compared to the non-linear dynamics has been shown in Figures 7.26 to 7.25. We use the initial conditions to initialise the integrator and the dry mass for the integrator to stop from Table 7.7. Figure 7.26, presents the trajectory from RK45 integrator with the nonlinear dynamics and linearised dynamics. Figure 7.23 and 7.24, present the individual components of position and velocity for both the integrators, respectively. As can be seen the integrator based on the linearised dynamics is verified to have the same trends as the one based on nonlinear dynamics. The overlay for the components of position and velocity are given in Figure 7.25. The difference in the two plots is in the range of 0.12 m for positions and 0.11 m/s for velocities, which is within the acceptance criteria from the mission requirements.

7.5.2. SCvx CQ Optimisation Algorithm

In this section, we verify the developed SCvx dynamic algorithm with ECOS in the MATLAB® interface with the test case presented in Szmuk et al. (2017). The SOCP formulation of the problem is present in Chapter 6

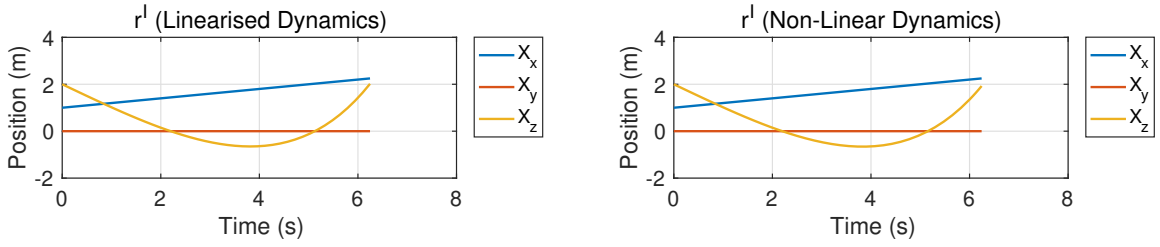


Figure 7.23: Simulated positions in the I frame.

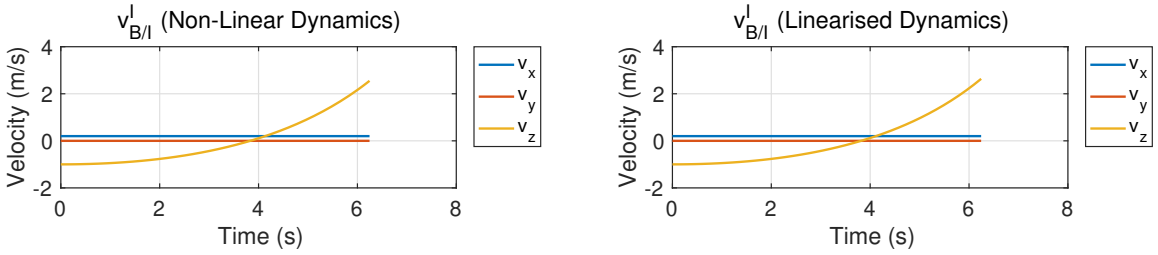


Figure 7.24: Simulated velocities in the I frame.

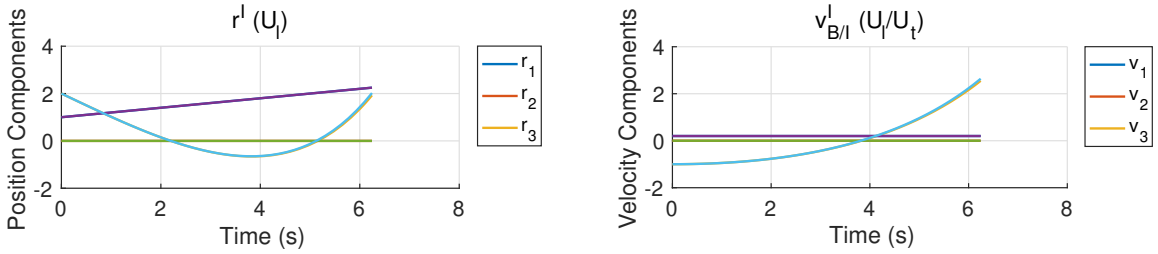


Figure 7.25: Overlay of position and velocity components obtained from the non-linear and linearised dynamics. The average difference in the two plots is 0.12 m for positions and 0.11 m/s for velocities.

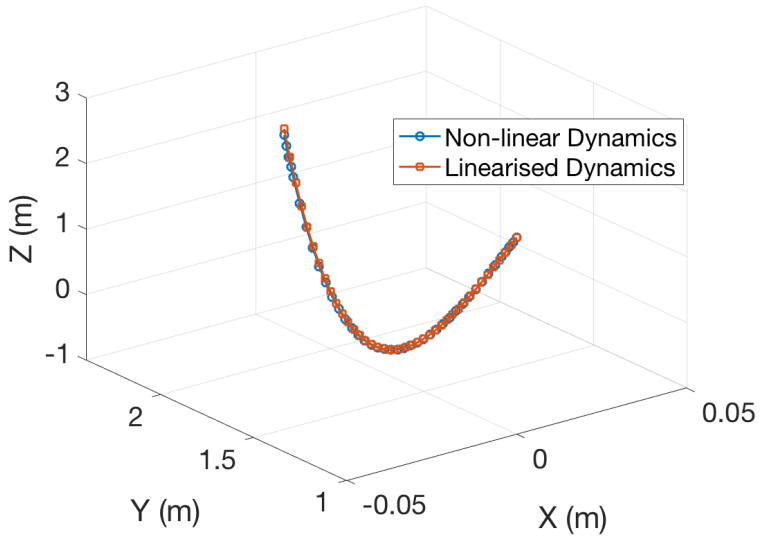


Figure 7.26: Simulated trajectories in the I frame for the non-linear dynamics integrator and linearised dynamics integrator.

and can be referred to for better interpretation of the results. Also the formulation for ECOS is briefly discussed in Section 7.2.3. Further details of the required parameters are mentioned in Table 7.9. The dynamic algorithm is run for 10 iterations and show convergence within the first 5 iterations. Figures 7.27 to 7.33 plot the converged parameters. A gradient of warm to cold is used for the convergence plots through all the results in this thesis.

Table 7.9: Additional Simulation Parameters for Descent Verification

K	i_{max}	T_{min} (N)	T_{max} (N)	δ_{max} (deg)	θ_{max} (deg)	γ_{min} (deg)	ω_{max} (deg/s)	w_v	w_{η_0}	σ	ς
30	10	0.5	3	10	20	10	30	145	1.22	3	1.2

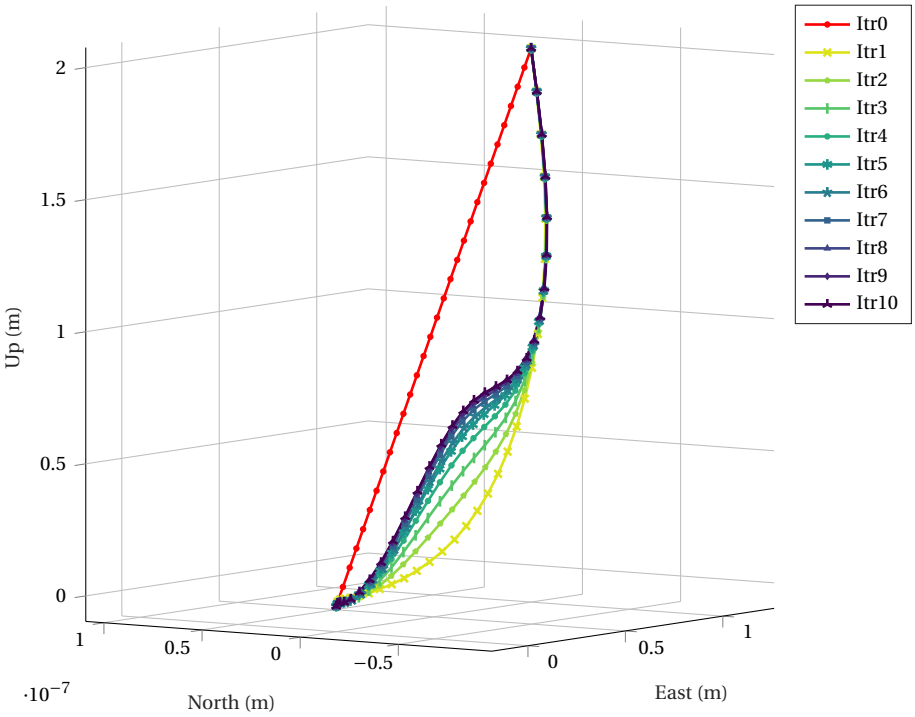


Figure 7.27: Trajectory convergence with SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

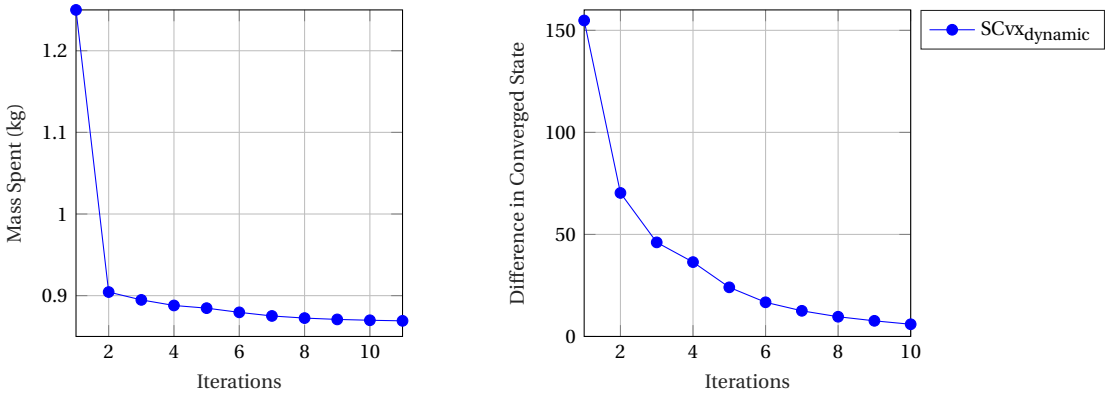


Figure 7.28: Spent mass (left) and difference between converged states (right)with SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

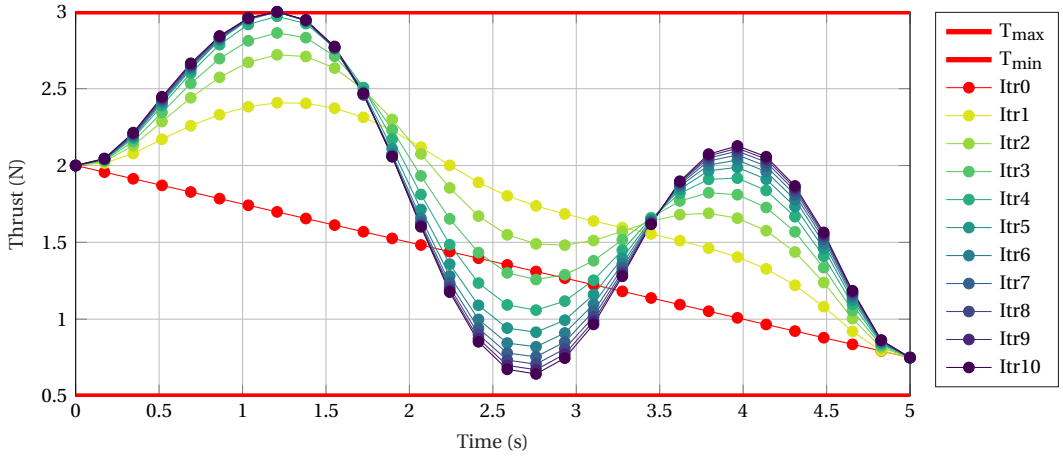


Figure 7.29: Thrust convergence with SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

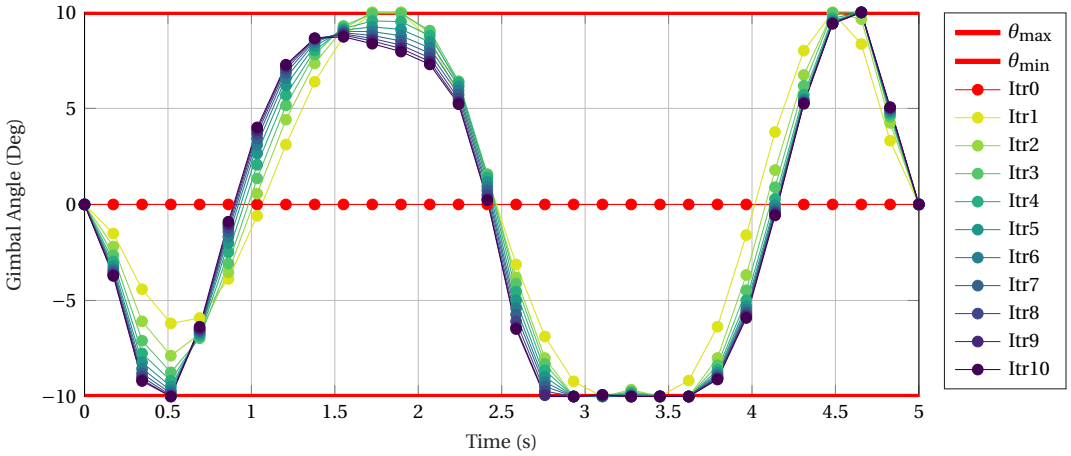


Figure 7.30: Gimbal angle convergence with SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

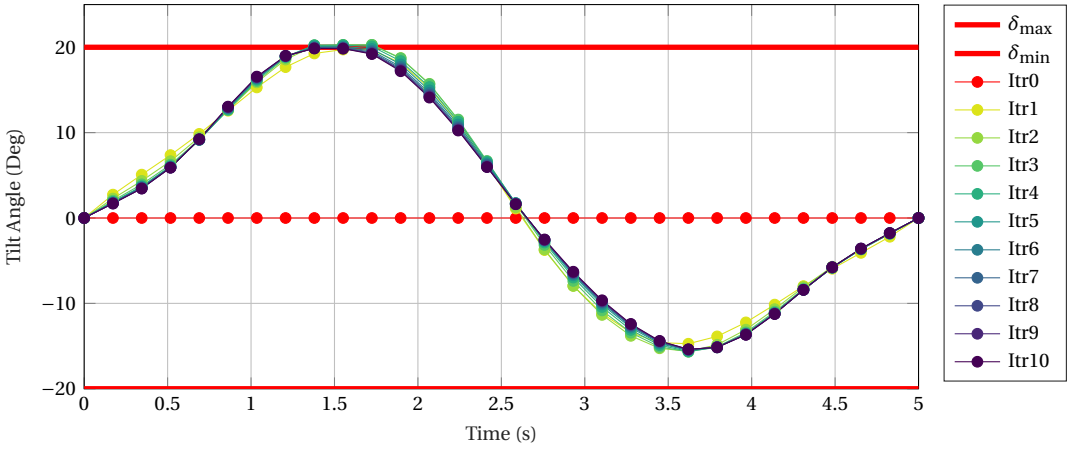


Figure 7.31: Tilt angle convergence with SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

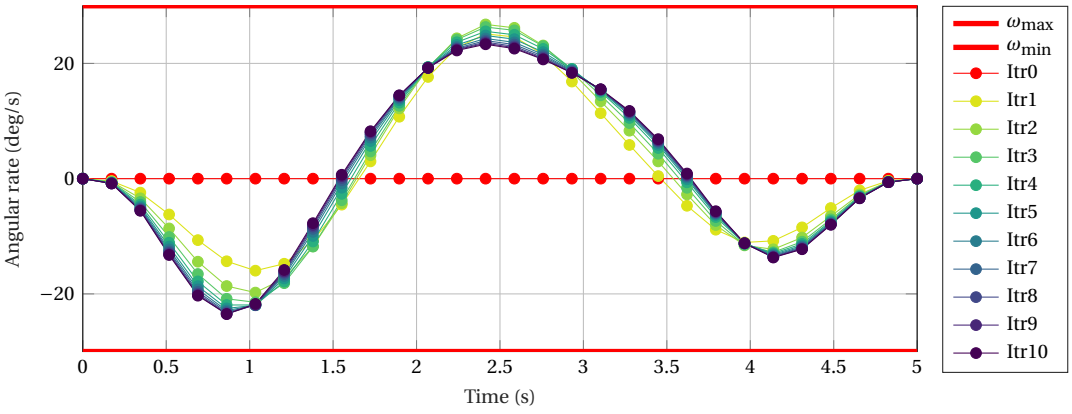


Figure 7.32: Angular rate convergence with SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

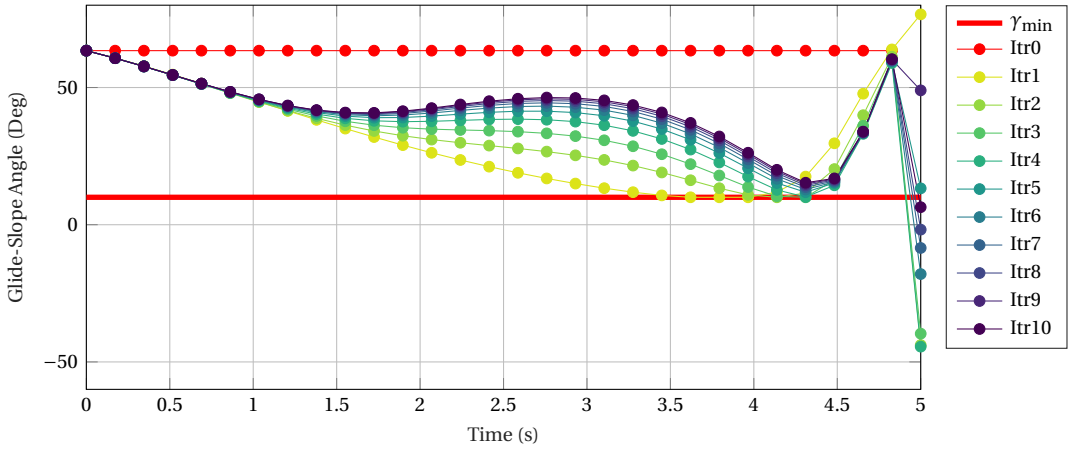


Figure 7.33: Glide-Slope angle convergence with SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

Table 7.10: Results from SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case.

Parameter	$\text{SCvx}_{\text{szmuk}}$	$\text{SCvx}_{\text{dynamic}}$	Unit
m_f	-	1.1316	kg
$\boldsymbol{r}_f^{\mathtt{I}}$	[0,0,0]	[0,0,0]	m
$\boldsymbol{v}_f^{\mathtt{I}}$	[-0.1,0,0]	[-0.1,0,0]	m/s
$\boldsymbol{\omega}_{\mathtt{B/I}_f}^{\mathtt{B}}$	[0,0,0]	[0,0,0]	rad/s
$\boldsymbol{q}_{\mathtt{B/I}_f}$	[0,0,0,1]	[0,0,0,1]	-
$\boldsymbol{T}_f^{\mathtt{B}}$	[0.75,0,0]	[0.75,0,0]	N
$\dot{\boldsymbol{T}}_f^{\mathtt{B}}$	[0,0,0]	[0,0,0]	N
θ_f	0	0	deg
δ_f	0	0	deg
γ_f	-	2.1138	deg

The results from the final iteration presented in Table 7.10 in comparison with the results from Szmuk et al. (2017). From this table it can be seen that the state for final time has been achieved with the SCvx dynamic algorithm. Figures 7.34 to 7.40 consist of results from the final iteration of the SCvx dynamic algorithm with the results from Szmuk et al. (2017) over the number of iterations or the time period of descent.

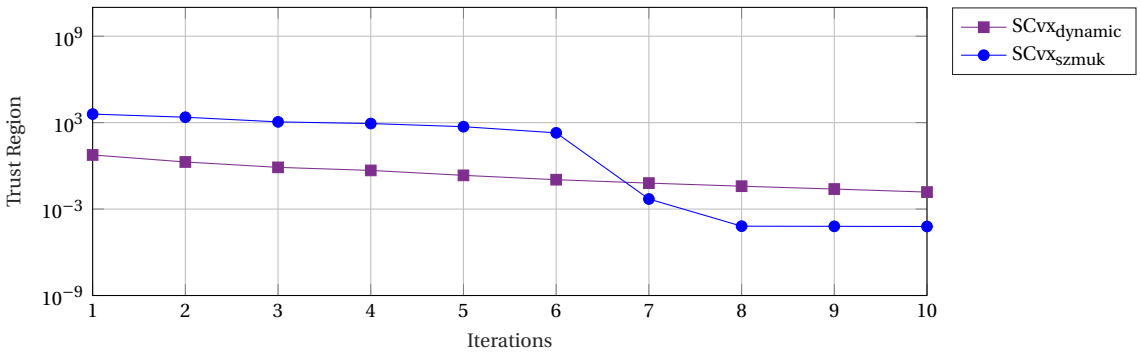


Figure 7.34: Trust regions for 10 iterations with SCvx dynamic CQ algorithm compared with the trust regions from Szmuk et al. (2017).

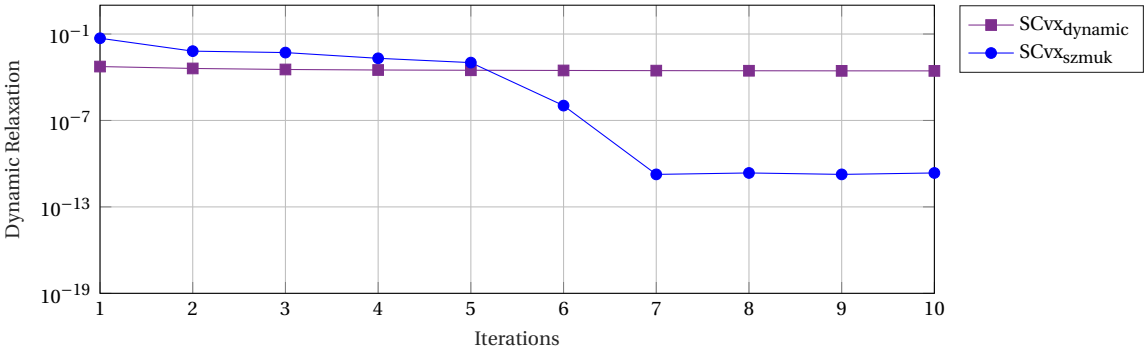


Figure 7.35: Dynamic relaxation for 10 iterations with SCvx dynamic CQ algorithm compared with the same from Szmuk et al. (2017).

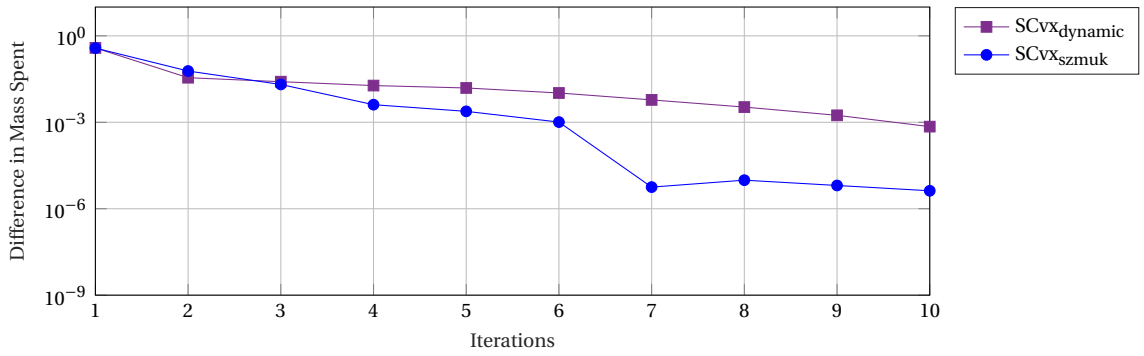


Figure 7.36: Convergence of mass in 10 iterations with SCvx dynamic CQ algorithm compared with the same from Szmuk et al. (2017).

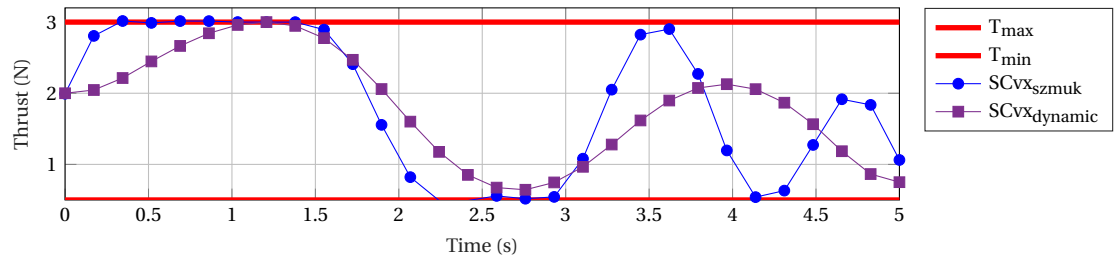


Figure 7.37: Thrust profile in the 10th iteration from the SCvx dynamic CQ algorithm run compared with the same from Szmuk et al. (2017).

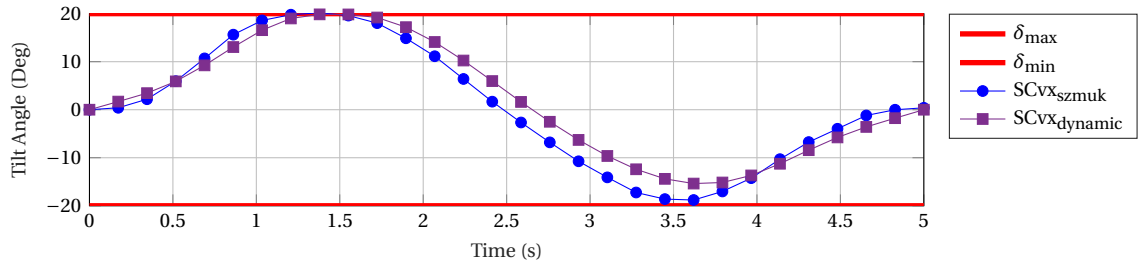


Figure 7.38: Tilt angle profile in the 10th iteration from the SCvx dynamic CQ algorithm run compared with the same from Szmuk et al. (2017).

The difference in the trust region and virtual controls (dynamic relaxation) trends is attributed to the dynamic nature of the developed algorithm. The update of trust region penalty weights (discussed in Section 6.3.6) determines the trust region radius values, and it follows a different logic than the one presented in Szmuk et al. (2017) and hence there is a difference. Similarly with the virtual controls, the implemented SOCP conversion implements two digressed Euclidean constraints to make it autonomous and hence we see

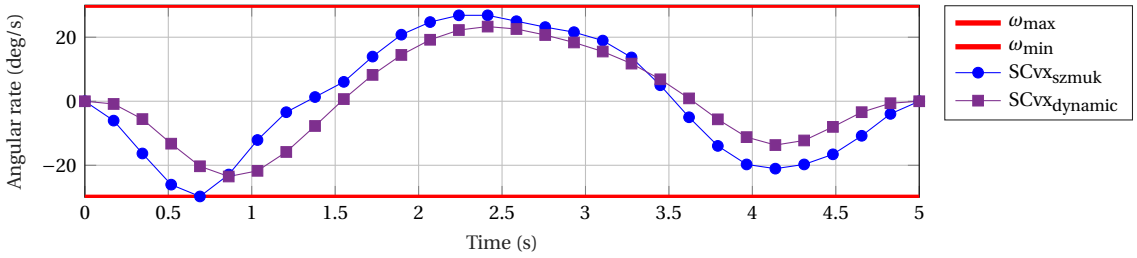


Figure 7.39: Angular rate profile in the 10^{th} iteration from the SCvx dynamic CQ algorithm run compared with the same from Szmuk et al. (2017).

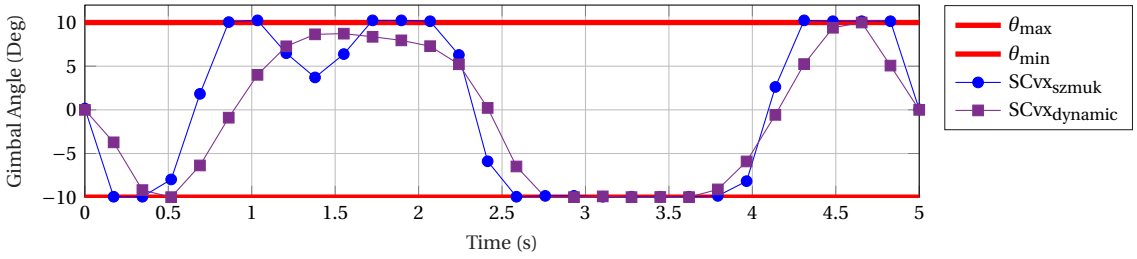


Figure 7.40: Gimbal angle profile in the 10^{th} iteration from the SCvx dynamic CQ algorithm run compared with the same from Szmuk et al. (2017).

a difference. However, since our mass convergence is in the range of 10^{-4} we consider the difference in the trust region and dynamic relaxation acceptable.

It can be seen from the Figures 7.37 to 7.40 that all the imposed constraints are met and the trends are similar. We also see smoother transitions in the angles as compared to the those obtained by Szmuk et al. (2017). However, they do not disclose the obtained final mass from their algorithm and hence we do not get a one to one verification. But the obtained thrusts from the SCvx dynamic algorithm are within the constraints and lower than Szmuk et al. (2017), which ultimately transcend to lower consumption of propellant.

A thorough reading of the Szmuk et al. (2017) paper shows a discrepancy in the attitude kinematic formula. It could be a typographic error, but we do get similar results as theirs in case of removal of the factor of 0.5 from the attitude kinematics. Another hiccup is the initialisation of the first derivative of thrusts as $\mathbf{0}$. This does not seem to be in sync with the thrust, since it is a product of the SC mass and the gravitational acceleration. We initialise the first derivative of thrusts with the mass dynamics. With this the unit test is passed and the algorithm considered to have been verified.

7.5.3. SCvx as Integrator

The controls solutions obtained from the above section are fed to the RK4[10] integrator to make sure the dynamic equations in the developed algorithm have been implemented correctly. The results are from the integrator has been plotted below, which generates the same trajectory as the optimiser thereby verifying the dynamics implemented in the SCvx algorithm. Figure 7.41 has the trajectory plotted from the RK4 integrator and the developed SCvx algorithm on the right and on the left is the second norm of the difference between the positions obtained from the two trajectories. As can be seen the difference is a maximum of 0.15 m, which is an acceptable error.

7.6. DQ 6DOF Asteroid Descent

In this section the pre requisite algorithms for the DQ optimiser have been verified. Compared to the CQ, we have two additional models that need to be verified to make sure the DQ SCvx optimiser works as required. Firstly, the initialisation of the of the DQ for each time step at iteration $i = 0$ needs to be a linear interpolation between the initial and final pose and attitude. Secondly, the polyhedral gravity of the asteroid to be implemented in the dynamics needs to be linearised separately due its highly non-linear relation with the pose and attitude DQ. Finally similar to the linearised 6DOF CQ dynamics, we need to verify the linearised

[10] RK4 has been used to integrate the trajectory with a fixed step, since the SCvx algorithm optimises the trajectory for fixed time steps

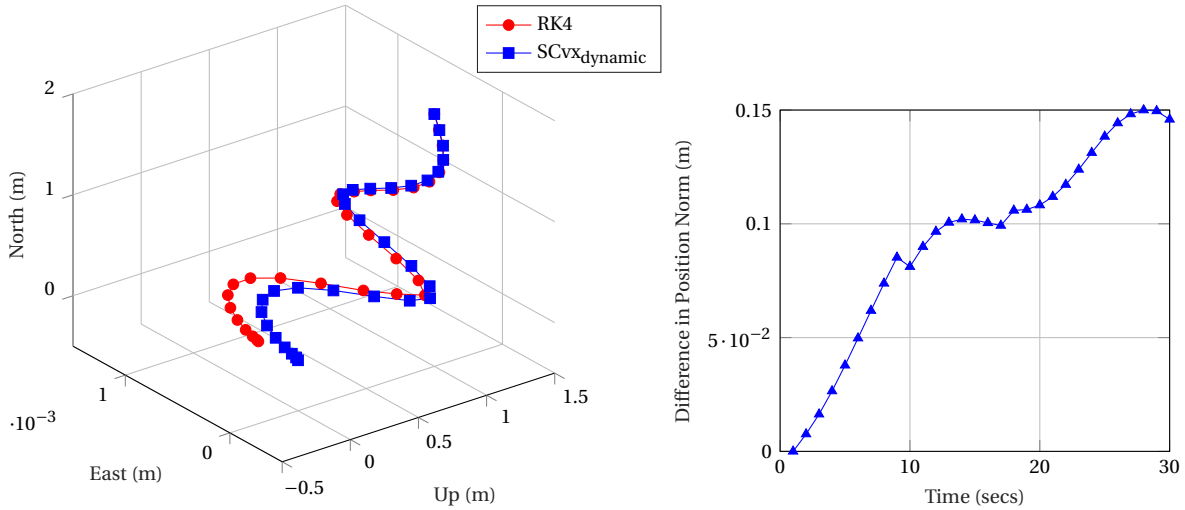


Figure 7.41: Comparison of RK4 integrator with SCvx dynamic algorithm results

DQ dynamics.

7.6.1. Complex-step Derivative Approximation of Polyhedral Gravity

Due to the complex and nonlinear relation of the polyhedral gravity field with the DQ, its partial derivative with respect to the DQ becomes too complicated and we need to use a numerical method. As discussed in Section 6.3.1 we have chose the complex-step derivative approximation to avoid subtraction errors of finite difference method. We present the partial derivative of the polyhedral gravity with respect to the DQ using different step sizes. We also perform a finite difference to make sure the results we get are consistent. A random position vector and quaternion are used to get a dual quaternion (Table 7.11) and then the gravity field at that point is generated with the polyhedral gravity field of Kleopatra. The results can be found in Table 7.12.

Table 7.11: Problem Parameters for Complex-Step Derivative

Parameter	Value
$\boldsymbol{r}^{\mathrm{A}}$ (m)	$[60e3, 70e3, 50e3]^{\mathrm{T}}$
$\boldsymbol{q}_{\mathrm{B/A}}$	$[0.3269, 0.5230, 0.5230, 0.5883]^{\mathrm{T}}$
$\check{\boldsymbol{q}}_{\mathrm{B/A}}$	$[0.3269, 0.5230, 0.5230, 0.5883, 1.242e4, 2.811e4, 1.046e4, -4.118e4]^{\mathrm{T}}$

The machine epsilon for MATLAB® is 10^{-16}, which means it can store upto 16 significant digits with accuracy. Therefore the results, till 10^8 are considered acceptable for the complex step derivative. The finite difference method is utilised to confirm that the results of complex step are indeed justified to give such low values. However, they are not used, due to them being prone to rounding of and truncation errors.

As can be seen from the results the partial derivative of the polyhedral gravity with respect to DQ is nearly zero. This makes sense, since the polyhedral gravity should not vary with the rotational quaternion and would have a substantial change with a change in km in the translational quaternion. However, since we do not have that kind of drastic changes in the position for the purpose of this thesis, we will consider the partial derivative of the polyhedral gravity with respect to the DQ to be zero. With the verification of SCvx algorithm with DQs in the Section 7.6.3 this assumption will be supported.

The complex partial derivative of the polyhedral gravity with respect to the DQ is independent of the resolution of the polyhedron structure of the asteroid. This is because the accuracy of the computed gravity varies with the structure, however, this does not affect its derivative with respect to the SC's attitude and position for the computed gravity with the utilised resolution.

Table 7.12: Partial derivative of Kleopatra's polyhedral gravity w.r.t. DQ with variable step sizes

h (Step Size)	Complex Differentiation	Finite Difference
0.1	[-11.915,-16.723,-8.760]	[-0.047,-0.0194,-0.0023]
0.01	[-0.097;-0.139,-0.108]	[-0.0011,-0.0005,0.0002]
10^{-3}	[-0.0003,-0.0004,-0.0007]	[-1.1e-05,-4.794e-06,1.9e-06]
10^{-4}	[-9.581e-07,-1.238e-06,-4.862e-06]	[-1.1e-07,-4.8e-08,1.9e-08]
10^{-5}	[-8.027e-09,-9.899e-09,-4.605e-08]	[-1.1e-09,-4.8e-10,1.9e-10]
10^{-6}	[-7.816e-11,-9.896e-11,-4.599e-10]	[-1.1e-11-4.8e-121.9e-12]
10^{-7}	[-7.837e-13,-9.996e-13,-4.604e-12]	[-1.1e-13,-4.8e-14,1.9e-14]
10^{-8}	[-7.840e-15,-1.0007e-14,-4.604e-14]	[-1.1e-15,-4.8e-16,1.9e-16]
10^{-9}	[-7.840e-17,-1.0008e-16,-4.604e-16]	[-1.1e-17,-4.8e-18,1.9e-18]
10^{-10}	[-7.840e-19,-1.0008e-18,-4.604e-18]	[-1.1e-19,-4.8e-20,1.9e-20]
10^{-11}	[-7.840e-21,-1.0008e-20,-4.604e-20]	[-1.1e-21,-4.8e-22,1.9e-22]

7.6.2. ScLERP Model

The SCvx optimiser has to be initialised as discussed in Section 6.3.8, and for the DQ representing the relative pose and attitude, we use ScLERP (Section 3.4.3) and this algorithm needs to be verified to generate a continuous screw motion, i.e., simultaneous rotation and translation. The test case we consider is given in Table 7.13. We convert them to an initial and final DQ, and perform the ScLERP operation for 40 nodes. Figure 7.42 shows the results from the ScLERP implementation and it can be seen that it provides with a linear interpolation between initial and final positions and angles.

Table 7.13: Test Parameters for ScLERP Model

Parameter	Initial	Final
$\check{\boldsymbol{q}}_{\mathrm{B/A}}$	$[0.3269, 0.523, 0.523, 0.5883, ..$	$[0.5866, 0.6599, 0.2933, 0.3666, ..$
	$12.4e3, 28.1e3, 10.5e3, -41.2e3]^{\mathrm{T}}$	$17.8e3, 3.1e3, 12.6e3 - 44.2e3]^{\mathrm{T}}$
$\boldsymbol{r}^{\mathrm{B}}$ (m)	$[60e3, 70e3, 50e3]^{\mathrm{T}}$	$[50e3, 65e3, 55e3]^{\mathrm{T}}$
θ (deg)	108	137

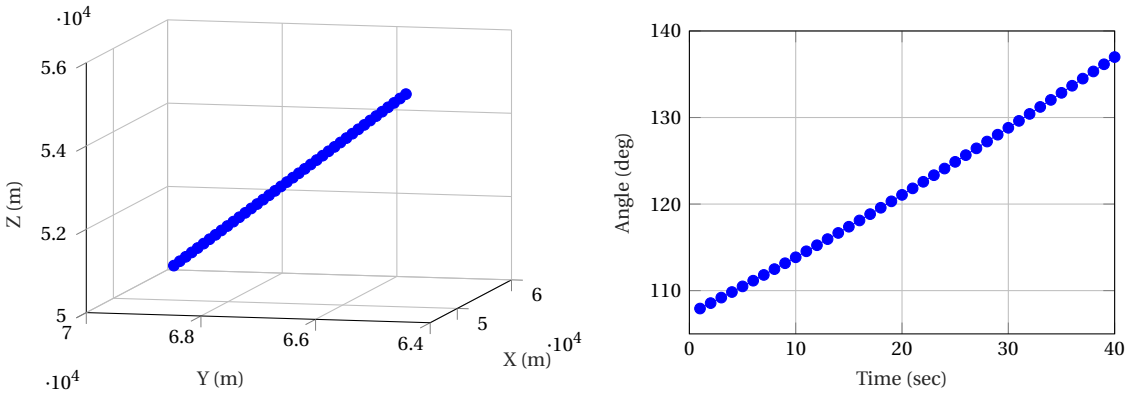


Figure 7.42: Results from ScLERP interpolation

7.6.3. SCvx DQ Optimisation Algorithm

In this section, we verify the SCvx dynamic algorithm using DQs instead of Cartesian coordinates and quaternions. We use the same test case as Szmuk et al. (2017). The test case is defined in Tables 7.7, 7.8 and 7.9. The only difference is the glide-slope constraint is switched off. Though the constraint has been derived for a DQ state, its form is quadratic with indefinite matrices. This makes it undesirable for conversion to SOCP. The only way to handle this constraint is linearisation and that has been left for further work, since it does not fit in the time frame of this thesis. The results from the DQ simulator and CQ simulator are presented in the Table 7.14 and in Figures 7.43 to 7.51 below. From Figure 7.43, it can be seen that the trajectory obtained by the SCvx CQ algorithm differs in the North axis in the range of 10^{-9} m, which is negligible and can be ignored.

Table 7.14: Results from SCvx dynamic DQ algorithm compared with results from the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

Parameter	SCvx_{CQ}	SCvx_{DQ}	Unit
m_f	1.119	1.107	kg
$\boldsymbol{r}_f^{\mathtt{I}}$	[0,0,0]	[0,0,0]	m
$\boldsymbol{v}_f^{\mathtt{I}}$	[-0.1,0,0]	[-0.1,0,0]	m/s
$\boldsymbol{\omega}_{\mathtt{B}/\mathtt{I}_f}^{\mathtt{B}}$	[0,0,0]	[0,0,0]	rad/s
$\boldsymbol{q}_{\mathtt{B}/\mathtt{I}_f}$	[0,0,0,1]	[0,0,0,1]	-
$\boldsymbol{T}_f^{\mathtt{B}}$	[0.75,0,0]	[0.75,0,0]	N
$\dot{\boldsymbol{T}}_f^{\mathtt{B}}$	[0,0,0]	[0,0,0]	N
θ_f	0	0	deg
δ_f	0	0	deg
γ_f	2.94	-5.91	deg

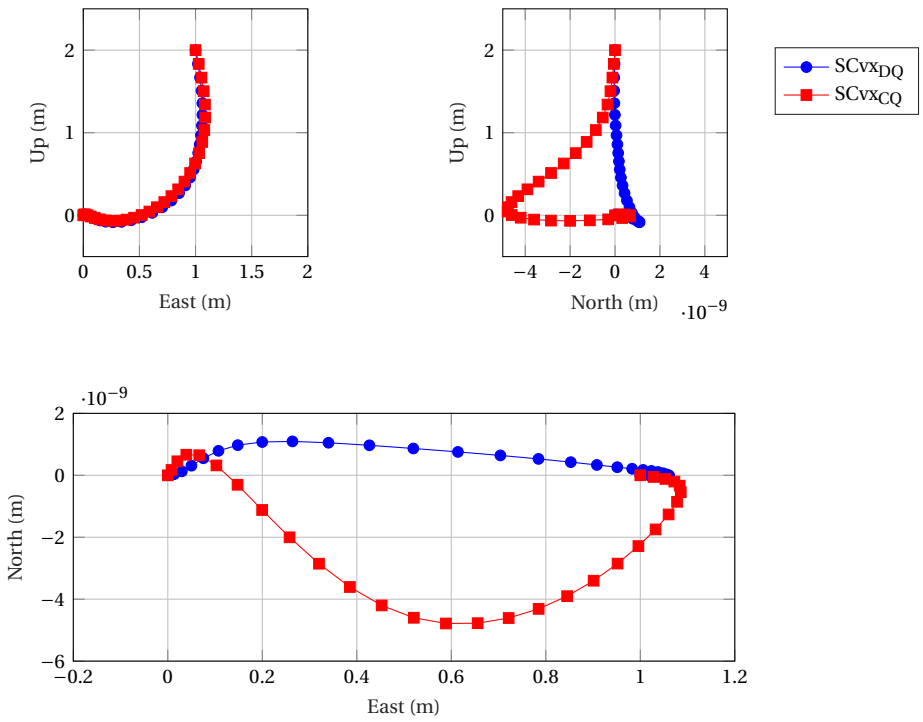


Figure 7.43: Trajectory (10^{th} iteration) comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

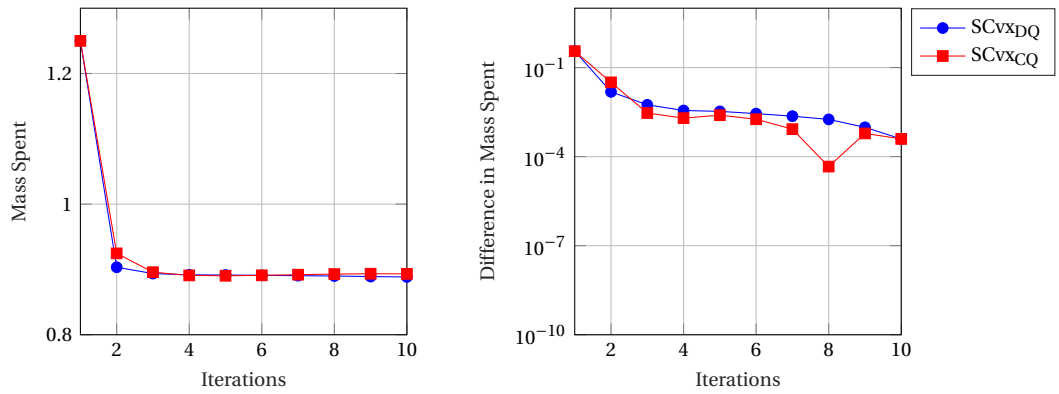


Figure 7.44: Spent mass and mass convergence comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

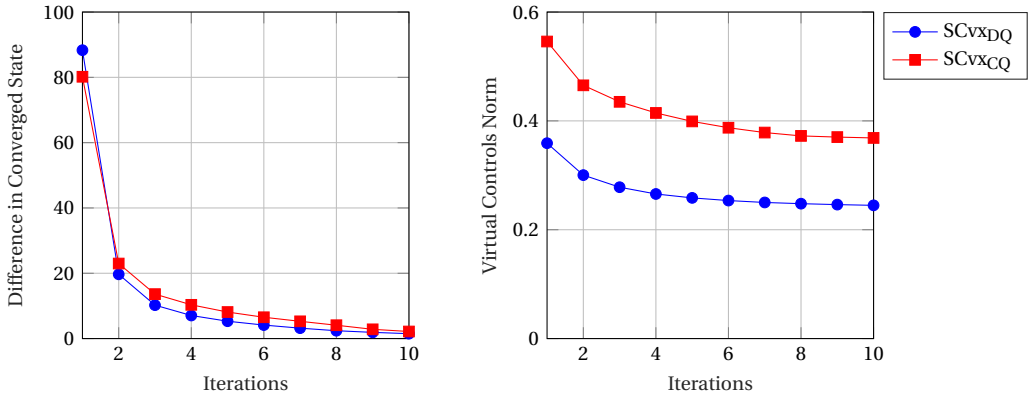


Figure 7.45: State convergence and virtual controls comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

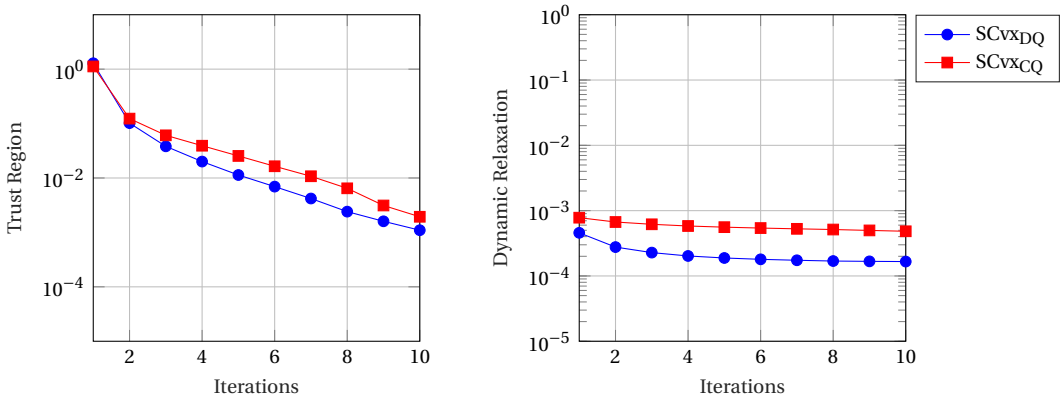


Figure 7.46: Trust region and dynamic relaxation comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

As seen from the results we get a lower mass consumption using the DQ algorithm. Figure 7.44 (right) shows the convergence in mass achieved after every iteration and as seen it follows the trend of the CQ algorithm closely. We see a better convergence in the state norm in Figure 7.45 (left) and lower virtual controls in the right. We also see a lower trend in trust region slack variable and the virtual control slack variable (dynamic relaxation) in Figure 7.46.

All the active constraints are met as shown in the Figures 7.47 to 7.51. The thrust and angular rate profile, plot the norm of the thrusts and angular rates, respectively for the duration of decent. The thrust profile and glide-slope angle trends closely follow the ones obtained from the SCvx CQ algorithm. The major difference is

seen in the angular rate, tilt angle and gimbal angle, however, since the constraints are met and DQ dynamics have been verified, these results are accepted.

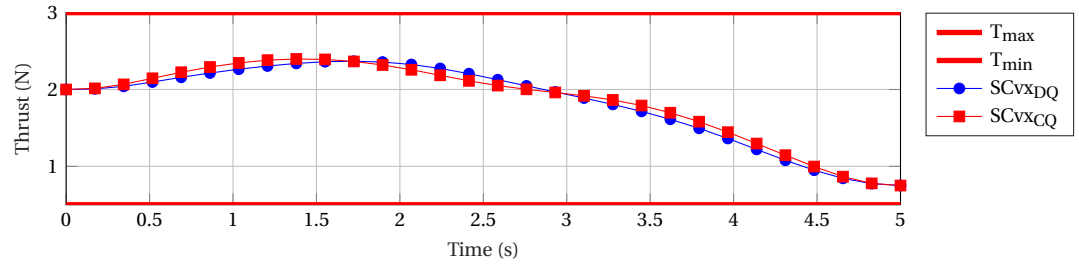


Figure 7.47: Thrust profile (10^{th} iteration) comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

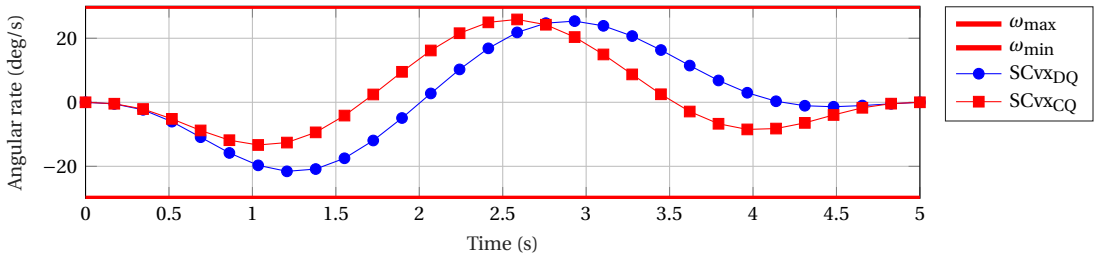


Figure 7.48: Angular rate (10^{th} iteration) comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

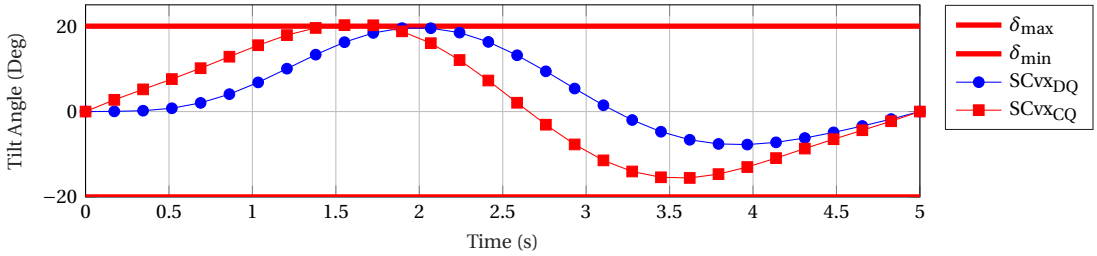


Figure 7.49: Tilt angle profile (10^{th} iteration) comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

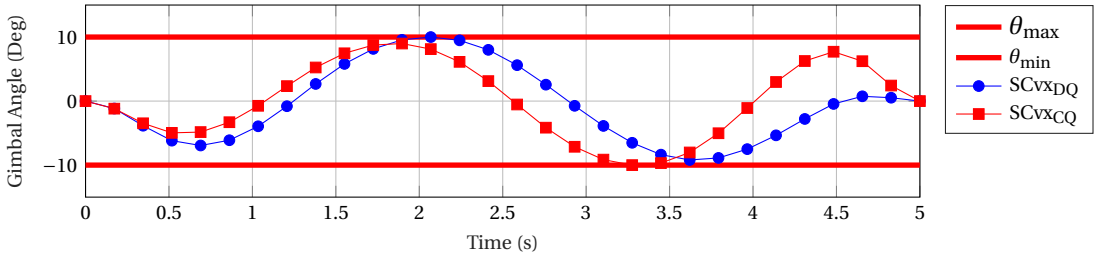


Figure 7.50: Gimbal angle (10^{th} iteration) comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

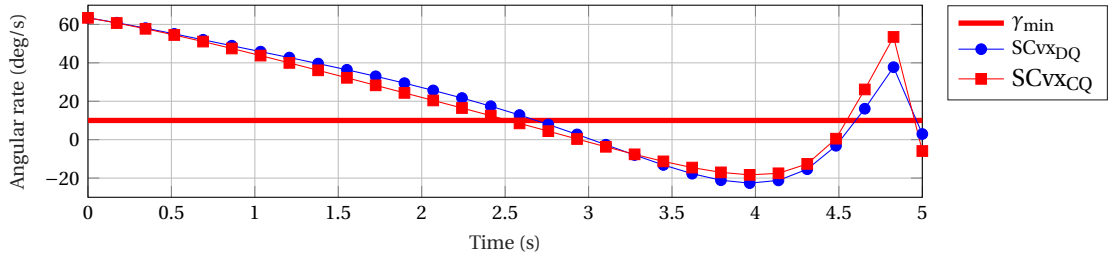


Figure 7.51: Glide-Slope angle (10^{th} iteration) comparison between SCvx dynamic DQ algorithm and the SCvx dynamic CQ algorithm for Szmuk et al. (2017) test case with glide slope constraint switched off.

7.7. Conclusion

In this chapter we have verified all the required models for the simulator. The SC models for mapping and descent have been illustrated and the asteroid models with polyhedron structure, illumination and landmarks have been put forth. The dynamics derived for the relative frame using dual quaternions has been verified. The external perturbance models have also been verified. The linearised dynamics for the Cartesian quaternion algorithm derived in Appendix B has been verified along with the developed dynamic successive convexification algorithm. The complex-step derivate of the polyhedral gravity field with respect to the relative frame dual quaternion has been derived as well as the ScLERP model for a linearly interpolated solution between dual quaternions has been implemented. Lastly, the dynamic successive convexification algorithm based on dual quaternions has been verified. These verification tests conclude that the developed algorithm for successive convex optimisation is working correctly and we can go ahead with the implementation of the chosen asteroids' dynamics and gravity field.

8

Simulations

In this chapter, we present the different outcomes from this thesis. The simulations are run for the asteroid Kleopatra, for the descent phase. The first section discusses the descent phase of the mission, where scenarios are built from scaling the Szmuk et al. (2017) test case, to understand the required penalty weights for the asteroid descent test case. Different scenarios have been considered for the descent problem in Section 8.1. The initialisation of the mapping problem can be found in Appendix A.

8.1. Descent Problem Scenarios

With previous experience from the verification of the SCvx algorithm, it has been found that obtaining an optimal solution for the descent problem depends not only on the feasibility of the test scenario at hand, but also the scale of the scenario and the chosen penalty weights. For this reason, different scenarios are first solved in the inertial frame, $\mathtt{I}$, using the SCvx dynamic algorithm with Cartesian coordinates and quaternions.

8.1.1. Scenario 1

The Mars 6DOF problem from Szmuk et al. (2017) has been scaled up by a factor of 10. The initial SC position, velocity, SC dry and wet mass and consecutively the SC inertia are multiplied by 10 from the Table 7.7 and 7.8. This also leads to increasing the limits on thrusts by a factor of 10. Tables 8.1 to 8.3 present the required parameters for the 1st scenario. The scaling parameters for the trust region penalty weight, σ and ς, are the same as mentioned in Table 7.9. The final time for the descent needs to be decided before hand. For a 20 m descent, with a unit acceleration due to gravity and initial torque of 20 N, a final time within the range 15 secs can be approximated by simple calculation. By trial and error, a time of 6 secs gives optimal trajectories, for the preset SCvx additional simulation parameters given in Table 8.3. The penalty weights for the virtual controls and trust regions, w_ν and w_η, are kept at the values of 10^4 and 1, to keep the virtual control values minimal and nominally penalise the trust region radius. The results from this scenario are presented in Figures 8.1 to 8.12 for the CQ dynamic algorithm.

Table 8.1: Algorithm Parameters for Scenario 1

$\boldsymbol{r}_0^{\mathtt{I}}$ (m)	$\boldsymbol{v}_0^{\mathtt{I}}$ (m/s)	$\boldsymbol{r}_f^{\mathtt{I}}$ (m)	$\boldsymbol{v}_f^{\mathtt{I}}$ (m/s)	$\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}_{0/f}}$ (deg/s)	$\boldsymbol{q}_{\mathtt{B/I}_{0/f}}$	m_{wet} (kg)	t_f (s)
$[10,0,20]^{\mathtt{T}}$	$[22,0,-10]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[0,0,-0.1]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$	20	6

Table 8.2: Simulation Parameters for Scenario 1

m_{dry} (kg)	J_0 (kg m^2)	$\boldsymbol{g}^I$ (m/s^2)	α (s/m)	$\boldsymbol{r}_{\mathtt{T}}^{\mathtt{B}}$ (m)
7.5	diag([5,5,5])	$[0,0,-1]^{\mathtt{T}}$	0.01	$[0,0,-1]^{\mathtt{T}}$

Table 8.3: Additional Simulation Parameters for Scenario 1

K	i_{max}	T_{min} (N)	T_{max} (N)	δ_{max} (deg)	θ_{max} (deg)	γ_{min} (deg)	ω_{max} (deg/s)	w_v	w_{η_0}
50	10	5	30	10	10	20	20	10000	1

Convergence is obtained in less than 10 iterations for the entire state and all constrained parameters. The total CPU time utilised by the optimiser for all iterations is ≈5 secs, whereas for the whole algorithm is ≈100 secs. Optimality is satisfied for each iteration of the SCvx dynamic algorithm. Details for each plot from Figure 8.1 to 8.14 are provided in the following paragraphs. Results obtained from this scenario are summarised in Table 8.4.

Figure 8.1 plots 3D trajectories for all iterations and they lie within the tolerance band of 10^{-2} m which is within the acceptable range for descent. The SC is seen to travel first along the Y-axis, which is due to thrust initially being towards in this direction to tilt the SC towards the landing site. This is a result of the inherent nature of the descent problem being non-minimum phase.

Figure 8.2 plots the spent mass and the state convergence obtained in 10 iterations. It can be clearly seen that convergence is achieved in just 4 iterations. Figure 8.4 provides a better visualisation in the convergence achieved for spent mass. Figure 8.3 plots the trust region radii, η, which can be seen to become numerically negligible after the fourth successive convexification iteration. Figure 8.5 follows a similar trend as seen in Figure 7.46. Also this value remains negligible through all iterations. These plots, therefore, help indicate convergence of the algorithm.

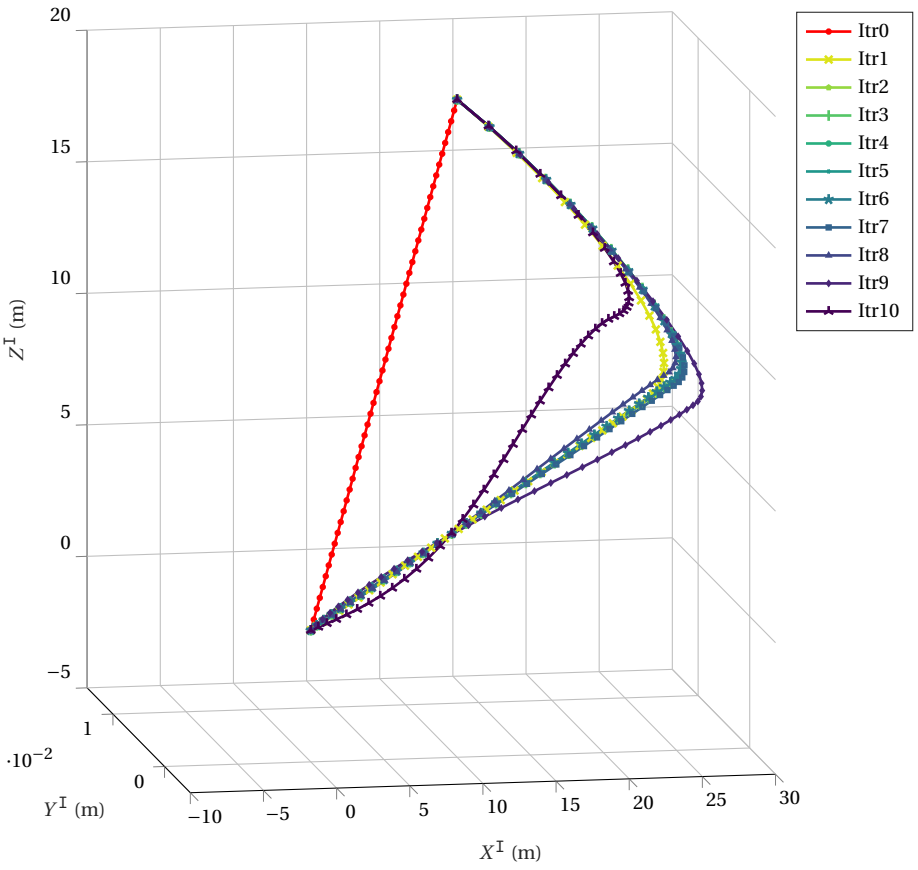


Figure 8.1: Trajectory convergence for Scenario 1

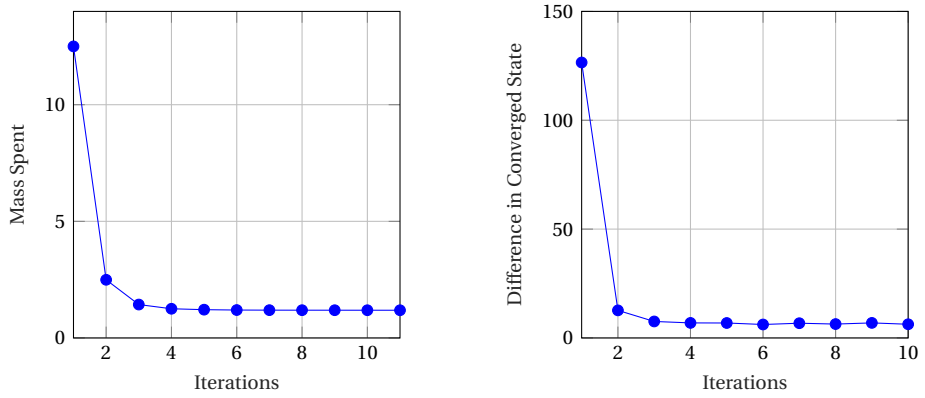


Figure 8.2: Spent mass (left) and difference between converged states (right) for Scenario 1

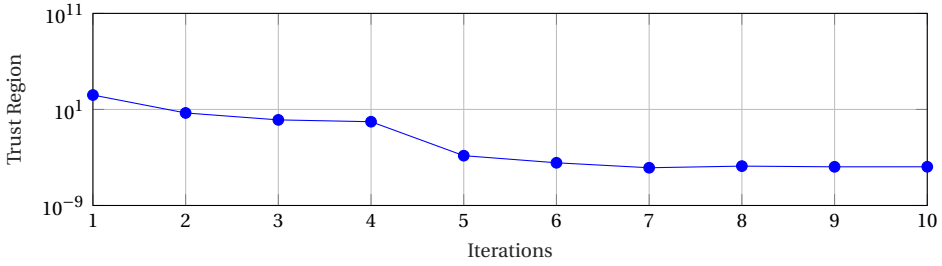


Figure 8.3: Trust region convergence for Scenario 1

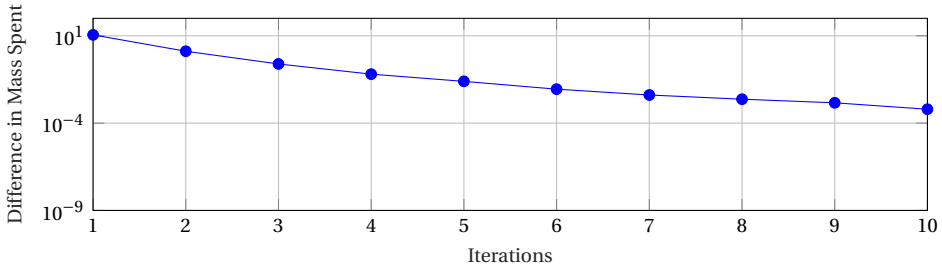


Figure 8.4: Mass convergence for Scenario 1

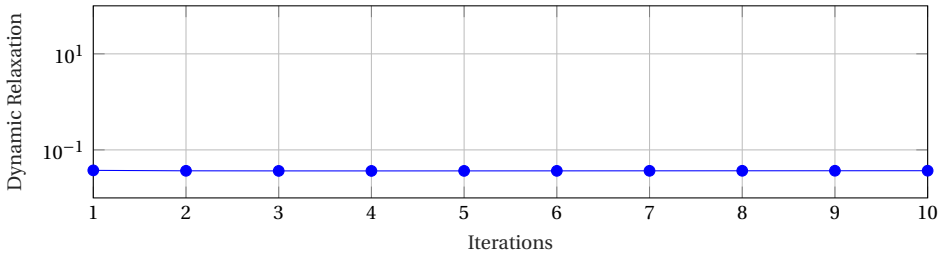


Figure 8.5: Dynamic relaxation convergence for Scenario 1

Figures 8.6 and 8.7 plot the norms of the thrusts and angular rates, respectively for the duration of descent for all 10 iterations. Figures 8.8, 8.9 and 8.10 plot the histories of the absolute vales of gimbal angles, tilt angles and glide-slope angles. It can be seen from these figures that the parameter optimisation problem activates all the constraints and follows them.

Figure 8.9 shows discrete jumps at the initial state and final nodes, since, these are not included in the dynamics and are boundary conditions to be satisfied by the optimiser. Note that the final values of the solution state are not included in the dynamics, since, there is no node after, for the problem to be discretised. The history of tilt angles follow that of gimbal angles and it can be seen at $t = 4$ sec, when the gimbal angle is minimal or rather the thrust direction is along the Z-axis of the body frame, the tilt angle of the SC also reduces. Figure 8.10 shows varied final values for the glide-slope angle, because here the constraint becomes ill-defined since the numerator and denominator both tend to go to zero.

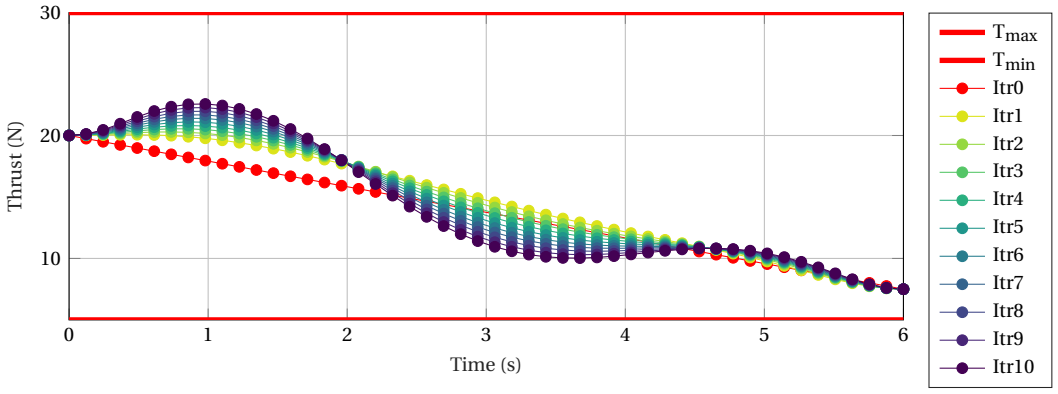


Figure 8.6: Thrust convergence for Scenario 1

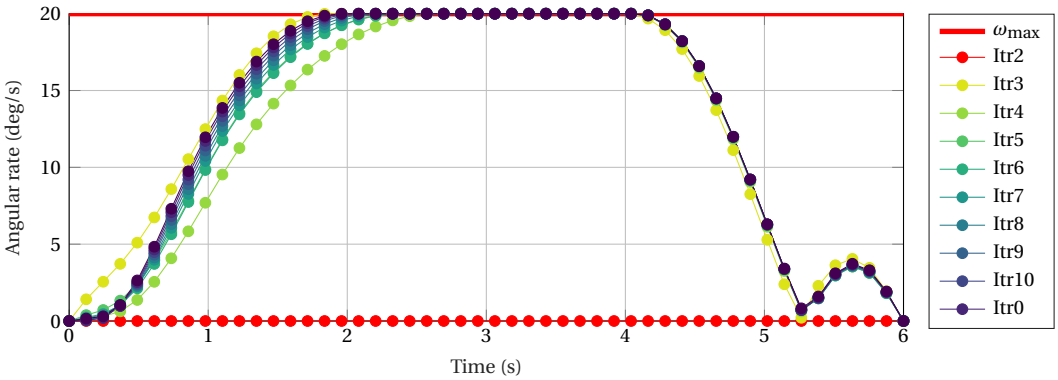


Figure 8.7: Angular rate convergence for Scenario 1

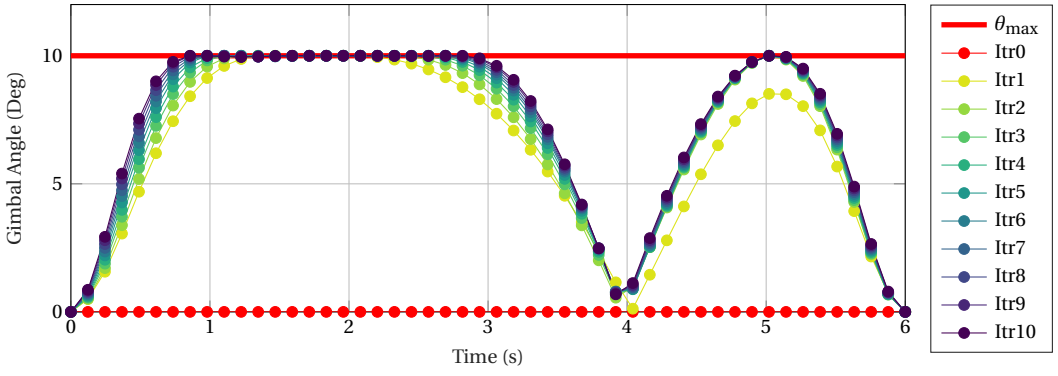


Figure 8.8: Gimbal angle convergence for Scenario 1

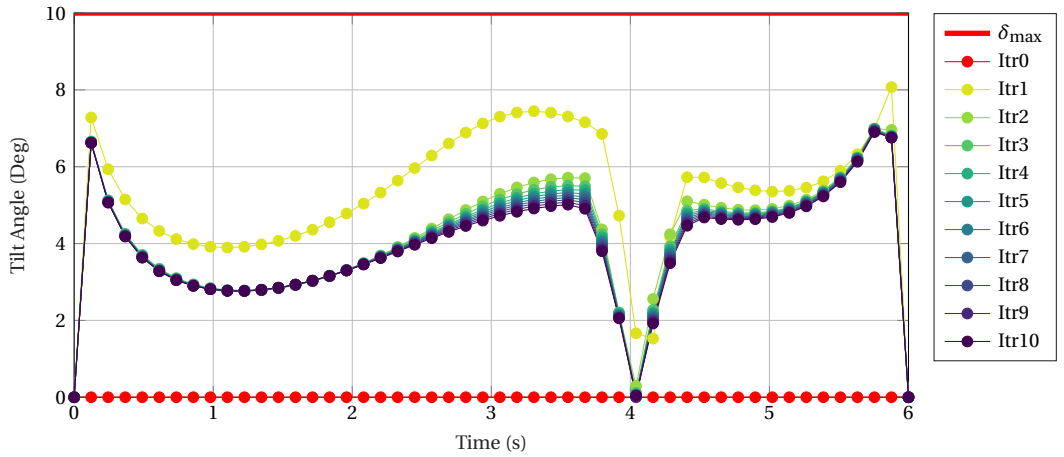


Figure 8.9: Tilt angle convergence for Scenario 1

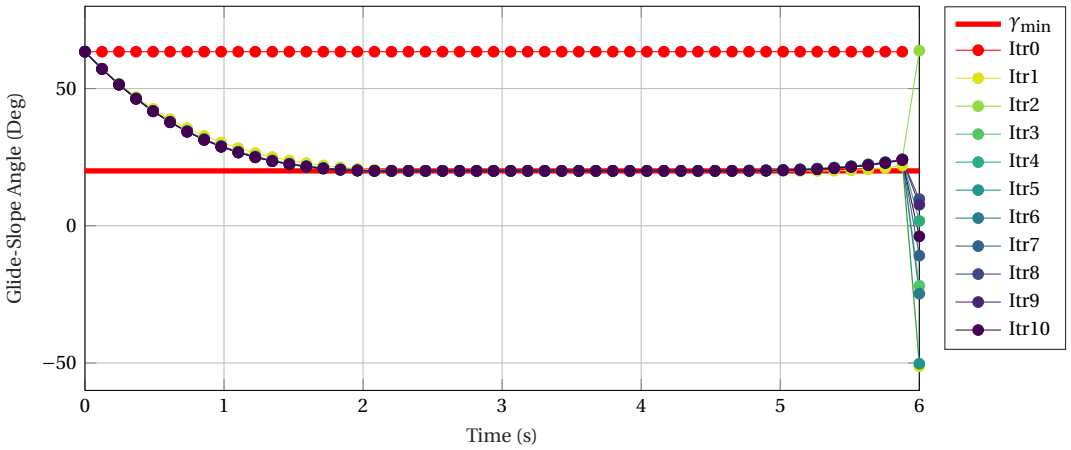


Figure 8.10: Glide-Slope angle convergence for Scenario 1

The thrust and angular histories for the 10^{th} iteration are plotted in Figures 8.11 to 8.14. There are no violations of the thrust lower boundary as seen for the Szmuk et al. (2017) test case, which happens due to the linearisation of the constraint. It can be seen from these plots that parameter optimisation problem does not violate the constraints, while activating them for consecutive time steps. This also proves that the discrete-time optimisation problem, stays faithful to the non-linear optimal control problem.

Table 8.4 puts forth the final state of the SC obtained after 10 iterations of iterative convexification. Unfortunately the end mass cannot be compared with any past missions at this stage, since a descent scenario minimally starts from the range of a few kilometres atleast. However, since all constraints are met and a converged optimal solution is obtained, this scenario is considered as a successful run for the algorithm.

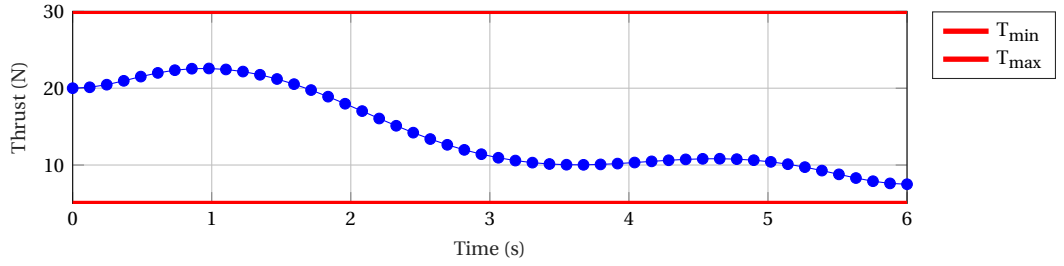


Figure 8.11: Thrust profile from the 10^{th} iteration for Scenario 1.

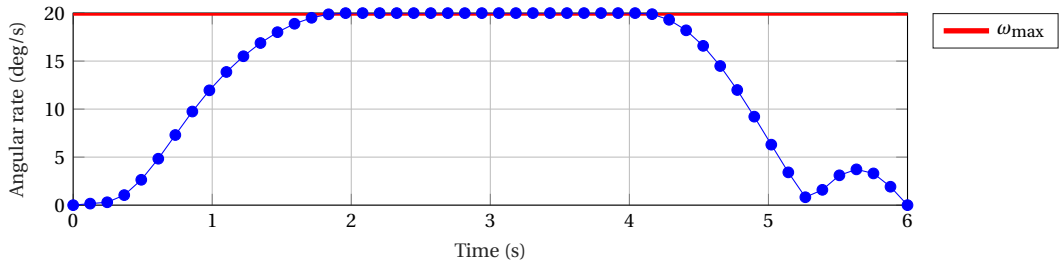


Figure 8.12: Angular rate profile from the 10th iteration for Scenario 1.

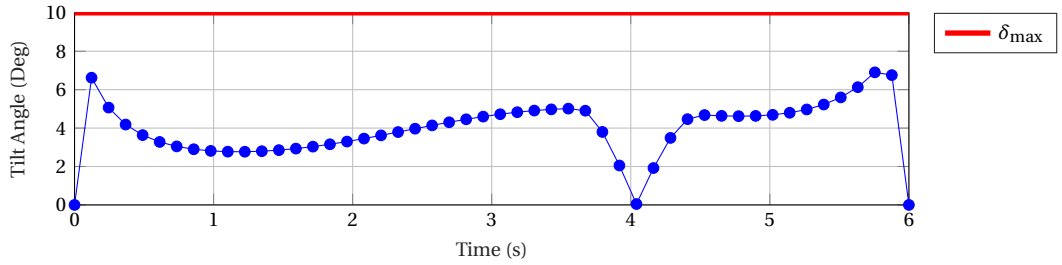


Figure 8.13: Tilt angle profile from the 10th iteration for Scenario 1..

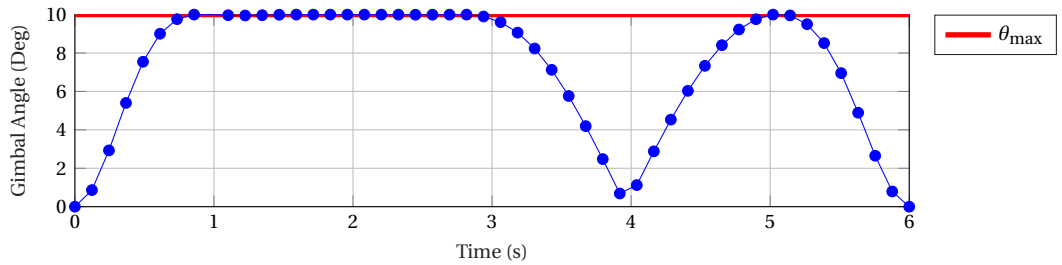


Figure 8.14: Gimbal angle profile from the 10th iteration for Scenario 1.

Table 8.4: Results for Scenario 1

Parameter	SCvx$_{dynamic}$	Unit	Parameter	SCvx$_{dynamic}$	Unit	Parameter	SCvx$_{dynamic}$	Unit
m_f	18.812	kg	$\boldsymbol{q}_{\mathtt{B/I}_f}$	[0,0,0,1]	-	δ_f	0	deg
$\boldsymbol{r}_f^{\mathtt{I}}$	[0,0,0]	m	$\boldsymbol{T}_f^{\mathtt{B}}$	[0,0,7.5]	N	γ_f	-3.86	deg
$\boldsymbol{v}_f^{\mathtt{I}}$	[0,0,-0.1]	m/s	$\dot{\boldsymbol{T}}_f^{\mathtt{B}}$	[0,0,0]	N/s			
$\boldsymbol{\omega}_{\mathtt{B/I}_f}^{\mathtt{B}}$	[0,0,0]	deg/s	θ_f	0	deg			

8.1.2. Scenario 2

Scenario 2 has been designed to be as close to a TAG descent problem as possible. The usual TAG descent problem as seen from the mission profiles of Hayabusa 2 and OSIRIS-REx, involve a descent from 100s of metres. Therefore, the initial position is increased as mentioned in Table 8.5. The initial velocity is increased to ≈ 30 m/s, which is approximately in the range of velocities for an SC around Kleopatra. The final velocity is kept at a cm/s, which is a mission requirement. The wet mass of the SC is 200 kg, which is still low as compared to the actual SC mass. However, this is kept so, since higher SC mass causes numerical problems for the solver. The inertia is still kept three-axes symmetrical, but 10 times that of Scenario 1. The maximum number of iterations are increased to 15, to ensure stability in the obtained results. The penalty weights on the trust region and virtual controls are still the same as Scenario 1, since higher values than those cause numerical problems for the solver. The final time is fixed to 5 secs, which is very low as compared to OSIRIS-REx and Haybusa 2 TAG descent profiles, but their target asteroids are Bennu and Ryugu, which are much smaller and have gravitational accelerations in the range of $\approx 10^{-5}$ m/s^2. The scaling parameters for the trust region penalty weight, σ and ς, are the same as mentioned in Table 7.9. The constant gravity is reduced to 5×10^{-2}, which is the expected gravitational acceleration at Kleopatra. All the parameters are summarised in the Tables 8.6 and 8.7.

Table 8.5: Algorithm Parameters for Scenario 2

$\boldsymbol{r}_0^{\mathtt{I}}$ (m)	$\boldsymbol{v}_0^{\mathtt{I}}$ (m/s)	$\boldsymbol{r}_f^{\mathtt{I}}$ (m)	$\boldsymbol{v}_f^{\mathtt{I}}$ (m/s)	$\boldsymbol{\omega}_{\mathtt{B}/\mathtt{I}_{0/f}}^{\mathtt{B}}$ (deg/s)	$\boldsymbol{q}_{\mathtt{B}/\mathtt{I}_{0/f}}$	m_{wet} (kg)	t_f (s)
$[100,0,200]^{\mathtt{T}}$	$[10,0,-30]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[0,0,-0.01]^{\mathtt{T}}$	$[0,0,0]^{\mathtt{T}}$	$[0,0,0,1]^{\mathtt{T}}$	200	5

Table 8.6: Simulation Parameters for Scenario 2

m_{dry} (kg)	J_0 (kg m^2)	$\boldsymbol{g}^I$ (m/s^2)	α (s/m)	$\boldsymbol{r}_{\mathtt{T}}^{\mathtt{B}}$ (m)
75	diag([50,50,50])	$[0,0,-0.05]^{\mathtt{T}}$	0.01	$[0,0,-1]^{\mathtt{T}}$

Table 8.7: Additional Simulation Parameters for Scenario 2

K	i_{max}	T_{min} (N)	T_{max} (N)	δ_{max} (deg)	θ_{max} (deg)	γ_{min} (deg)	ω_{max} (deg/s)	w_ν	w_{η_0}
50	25	2	30	10	20	20	10	10000	1

Convergence is obtained in less than 6 iterations for the entire state and all constrained parameters. The total CPU time utilised by the optimiser for all iterations is ≈7 secs, whereas for the whole algorithm is ≈140 secs. Optimality is satisfied for each iteration of the SCvx dynamic algorithm. Figure 8.15 to 8.30 are explained in detail in the following paragraphs. Table 8.8 holds the results obtained from this scenario.

Figure 8.15 plots 3D trajectories for all iterations and they lie within the tolerance band of 10^{-5} m of each other, which is well within the acceptable range for convergence. We see a similar profile as with Scenario 1 for the SC travel. It can be seen to travel along the Y-axis, as the thrusts tilt the SC towards the landing site and then it readjust the thrusts to come to the descent location. As mentioned earlier this a consequence of the non-minimum phase nature of the problem.

Figure 8.16 plots the spent mass and the state convergence obtained in 10 iterations. Figure 8.19 shows that a convergence of 10^{-2} is achieved after a drop at the 6$^{\text{th}}$ iteration, where the optimiser manages to over predict a lower cost, but stabilises after the next 4 iterations. Comparing it with Scenario 1, Figure 8.4, it can be seen that a lower convergence is achieved. Figure 8.17 shows η becoming numerically negligible right after the first iteration and stabilising after iteration 5. However, from Figure 8.16, it can also be seen that the convergence of the state is achieved at a higher value than Scenario 1. Figure 8.18, that plots ν for all the iterations does not change by much and remains at 10^1 with convergence. This is a problem, since it indicates that the optimal control problem still relies on virtual controls to find an optimal solution. This asks for a higher penalty weight on the slack variable used for the virtual controls in the cost function. However, this option is limited by the optimiser, being unable to handle higher values, due to a badly scaled problem.

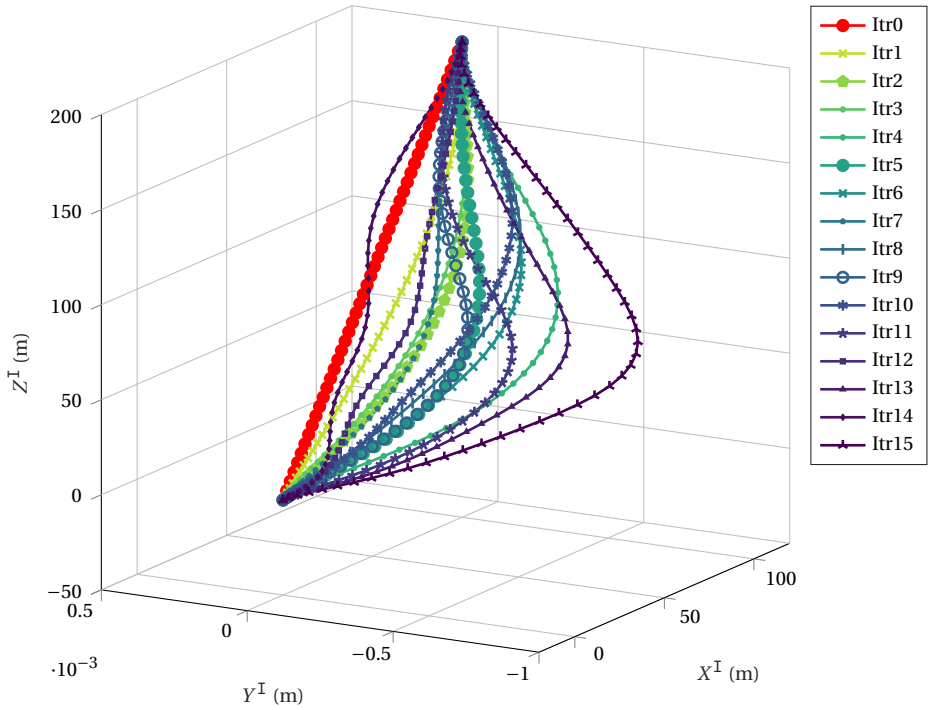


Figure 8.15: Trajectory convergence for Scenario 2.

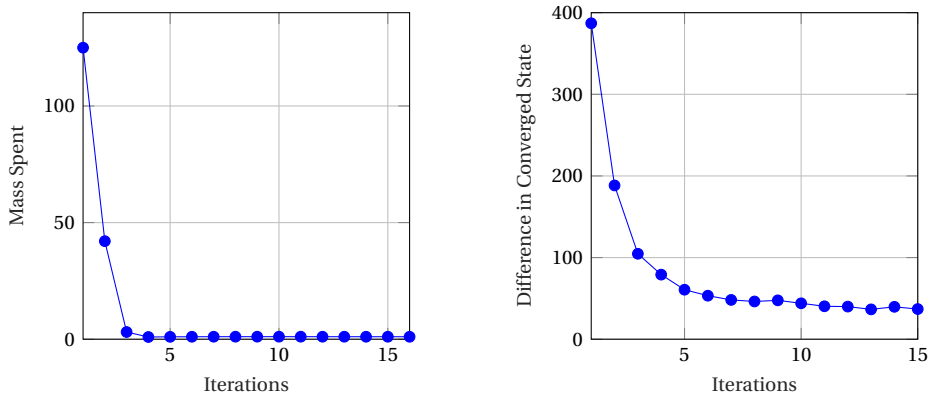


Figure 8.16: Spent mass (left) and difference between converged states (right) for Scenario 2

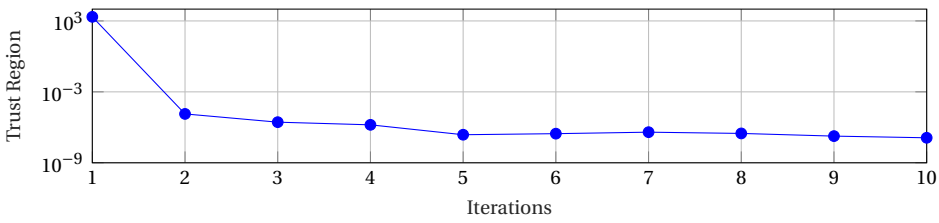


Figure 8.17: Trust region convergence for Scenario 2

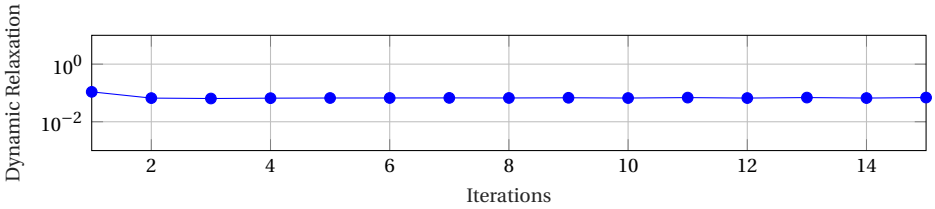


Figure 8.18: Dynamic relaxation convergence for Scenario 2

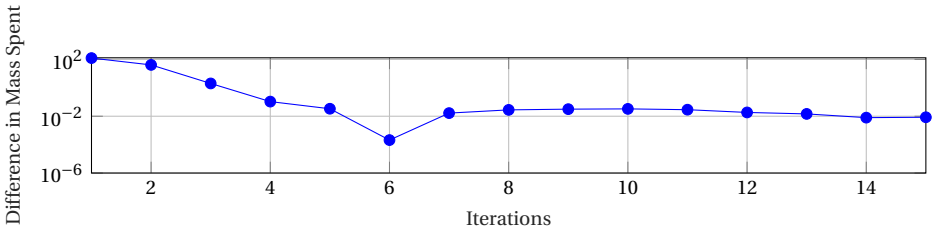


Figure 8.19: Mass convergence for Scenario 2

Figures 8.20 is the convergence plot of the norm of the thrusts over the descent duration. The thrust profile is seen to increase gradually after 0.5 secs. The velocity of the SC increases in the negative Z-direction till 2 secs, to let the SC cover a major distance in a short time after which the thruster is activated to maximum, and the velocity is reduced to a cm/s at final time. This can be seen in Figure 8.23, which plots the velocity profiles for all the iterations. Also as seen from the glide-slope angles in Figure 8.26, they are much higher than the required constrained angle from the initial position itself. This transcends to no torque required initially for the SC to be moved such that it travels inside the allowed cone.

Figure 8.21 plots the angular rate norm for all 15 iterations. It can be seen that the constraint on the angular rate is activated. However, a clear convergence is not obtained. This is due to the individual Y-axis thrust component varying with every iteration. Similar to Scenario 1, the tilt angles shown in Figure 8.25 show discrete jumps at the initial state and final nodes, since, these are not included in the dynamics and are boundary conditions to be satisfied by the optimiser. Also, it can be seen at $t = 2.85$ sec, where the gimbal angle is minimum (Figure 8.24) and the thrust direction is along the Z-axis of the body frame, the tilt angle of the SC is reduced to zero as well. Figure 8.10 shows the glide-slope angle remaining above 40°, throughout the descent profile, since it begins with 60°. This proves that the solution remains within the cone of glide-slope constraint during the descent while utilising this high angle as a benefit for faster descent.

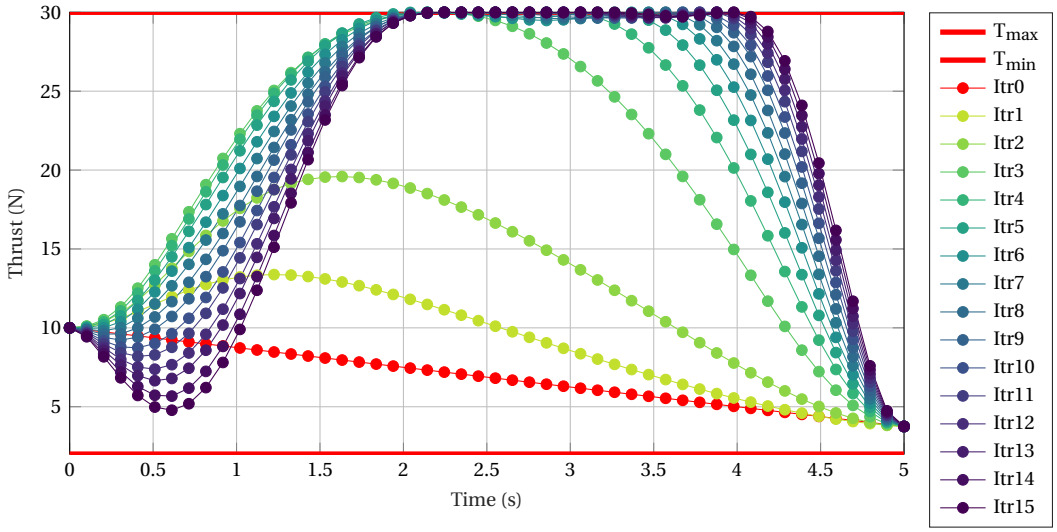


Figure 8.20: Thrust convergence for Scenario 2

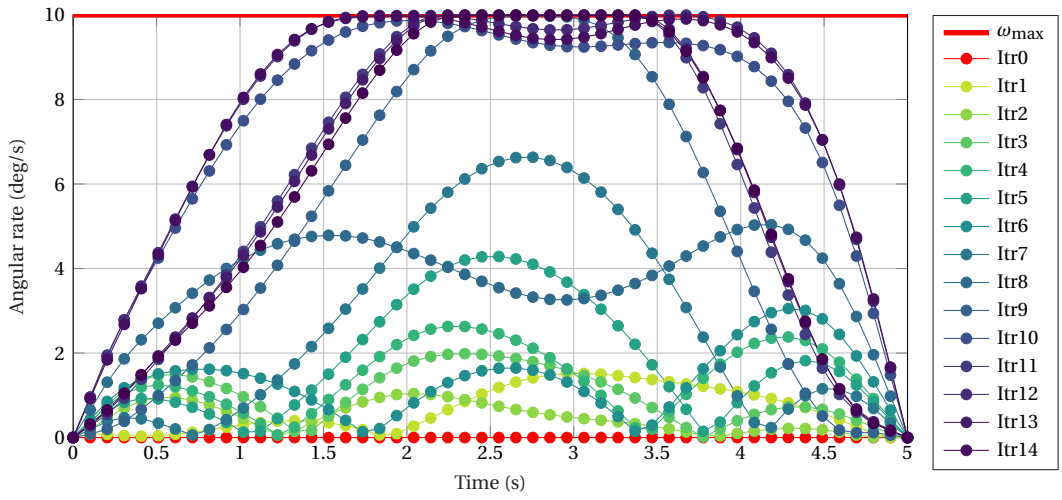


Figure 8.21: Angular rate convergence for Scenario 2

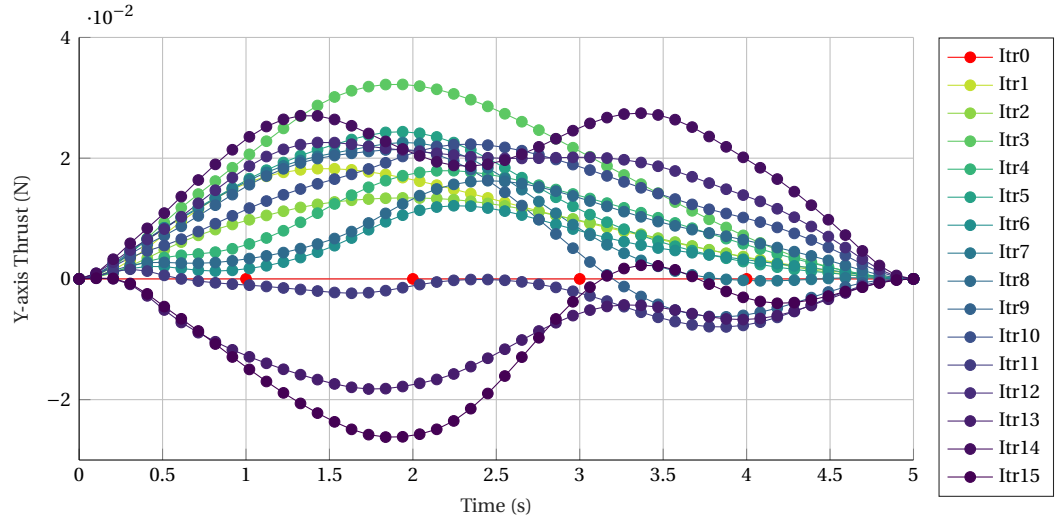


Figure 8.22: Y-axis components of thrust for all iterations of Scenario 2

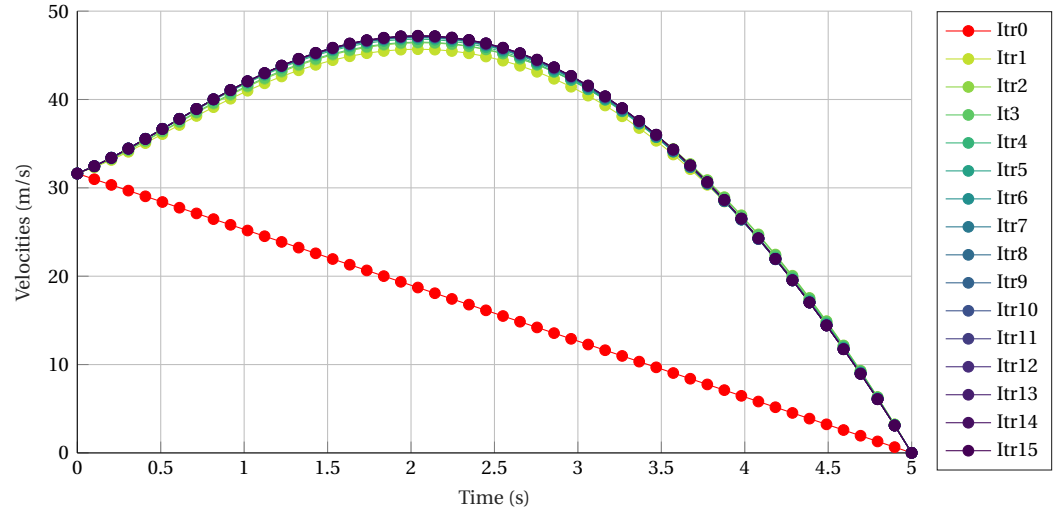


Figure 8.23: Velocity profiles for all iterations for Scenario 2

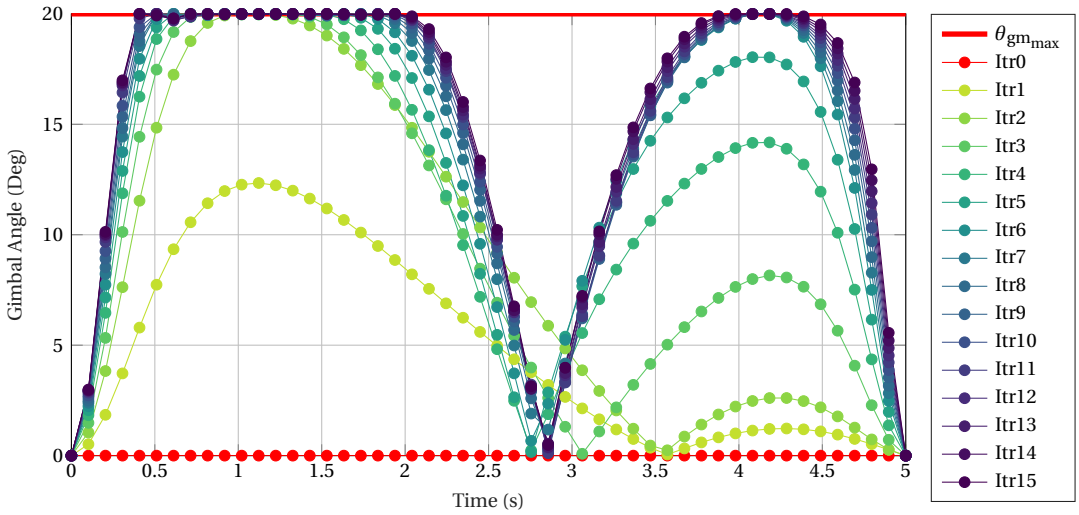


Figure 8.24: Gimbal angle convergence for Scenario 2

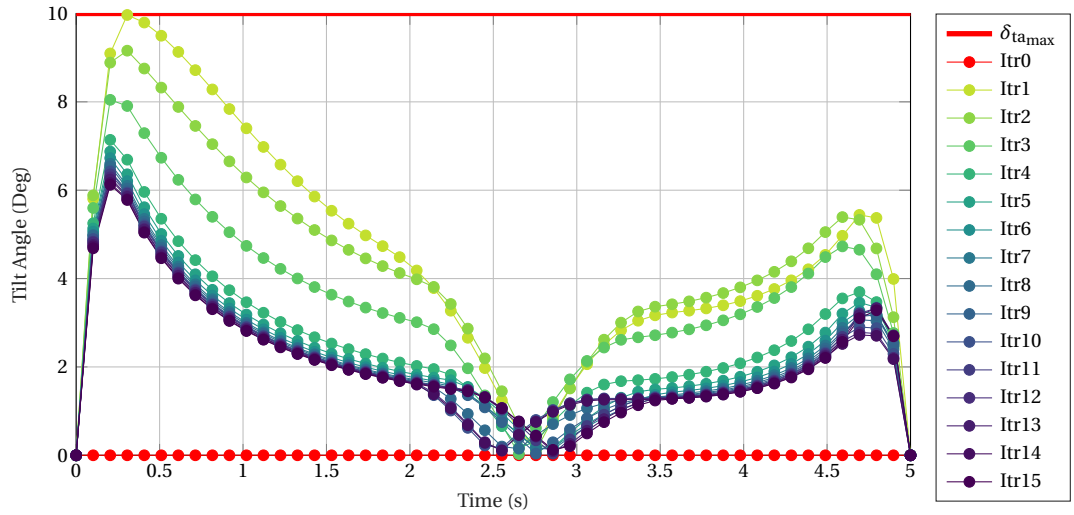


Figure 8.25: Tilt angle convergence for Scenario 2

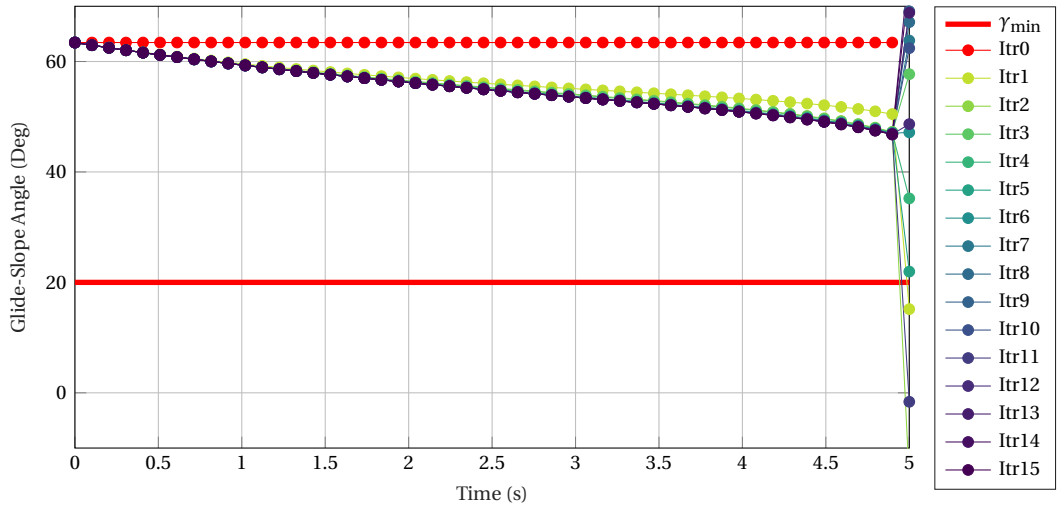


Figure 8.26: Glide-slope angle convergence for Scenario 2

The thrust and angular histories from the 15$^{\text{th}}$ iteration are plotted in Figures 8.27 to 8.30. No violations are seen for all the active constraints. A requirement of maximum thrust is seen for ≈40% of the duration of descent. This could be avoided by longer duration of descent with shorter thrust burns. But as conveyed earlier, the optimiser faces numerical issues while handling a longer duration. However, the results prove that the parameterisation problem has been correctly implemented, but needs to be tuned to satisfy mission profiles as per the requirement.

Table 8.8 holds the final state of the SC after 15 iterative convexifications. A propellant mass of 1.0568 kg is consumed at the final iteration. Due to the short duration of flight the propellant usage is low and it increases with an increase in final time for this particular scenario. However, since all constraints are met and a converged optimal solution is obtained, this scenario is considered as a successful run for the algorithm. Although by convergence, the discrete-time problem stays faithful to the non-linear problem, the scenario is not applicable in real time, due to its low duration. Usually TAG descent profiles last for 10-15 mins.

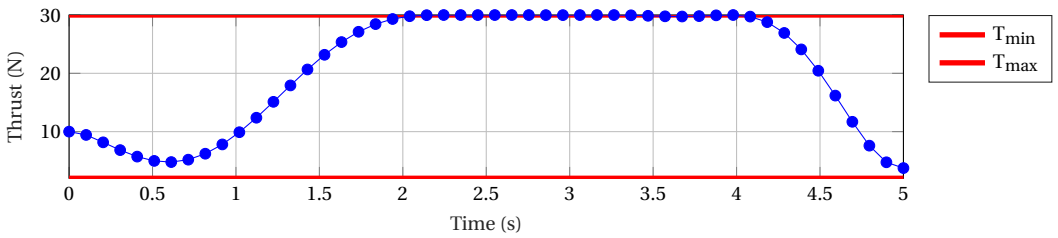


Figure 8.27: Thrust profile from the 15^{th} for Scenario 2.

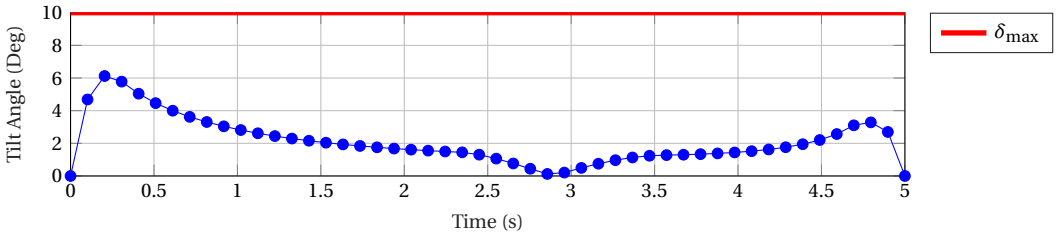


Figure 8.28: Tilt angle profile from the 15^{th} for Scenario 2.

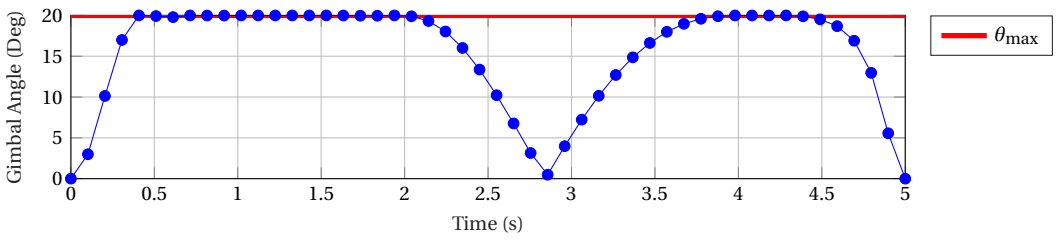


Figure 8.29: Gimbal angle profile from the 15^{th} for Scenario 2.

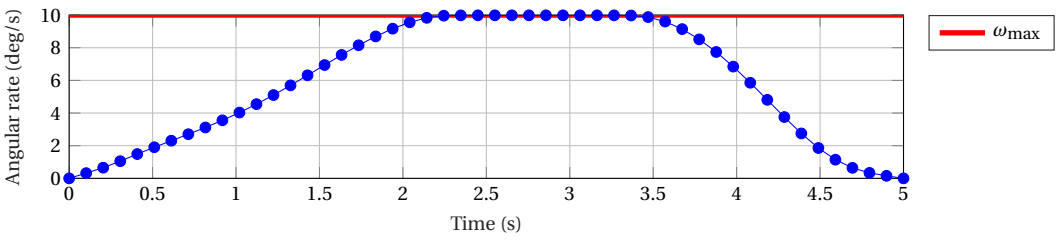


Figure 8.30: Angular rate profile from the 15^{th} for Scenario 2.

Table 8.8: Results for Scenario 2

Parameter	$SCvx_{dynamic}$	Unit	Parameter	$SCvx_{dynamic}$	Unit	Parameter	$SCvx_{dynamic}$	Unit
m_f	198.94	kg	$\boldsymbol{q}_{\mathtt{B}/\mathtt{I}_f}$	[0,0,0,1]	-	δ_f	0	deg
$\boldsymbol{r}_f^{\mathtt{I}}$	[0,0,0]	m	$\boldsymbol{T}_f^{\mathtt{B}}$	[3.75,0,0]	N	γ_f	74.54	deg
$\boldsymbol{v}_f^{\mathtt{I}}$	[0,0,-0.01]	m/s	$\dot{\boldsymbol{T}}_f^{\mathtt{B}}$	[0,0,0]	N/s			
$\boldsymbol{\omega}_{\mathtt{B}/\mathtt{I}_f}^{\mathtt{B}}$	[0,0,0]	deg/s	θ_f	0	deg			

8.2. Test Runs

The results from Scenario 2 as discussed above already show sensitivity to numerical scaling of the parameters. The author has investigated numerous scenarios with varying parameters. The best two scenarios of the test runs have been put forth in the thesis. There are a few observations from the various test runs conducted,

- Scaling up the SC initial state to replicate an actual mission profile, where the initial positions would be in kilometres for an asteroid the size of Kleopatra, increasing the SC mass to 1000s of kgs and using a non axis-symmetric inertia and using a specific impulse greater than 2000 s^{-1} renders the problem infeasible or numerically incomprehensible for the optimiser. Various combinations of the above parameters have been tried and tested.

- The penalty weights that scale the virtual controls and trust region slack variables in the cost function are not logically derivable. Their relationship with the scale of the problem is not straightforward and a number of tries have been made to find out the best possible values for the particular scenarios presented above.

- Finding an optimal solution for the problem is sensitive to the descent time. Higher descent times for the above scenarios have been tried, but they do not meet the tolerance for optimality after a few iterations of convexification or run into numerical problems due to the dynamic nature of the penalty weight on the trust region slack and sometimes even run into infeasibility after optimally solving a few iterations.

Since both Scenario 1 and 2 are feasible and provide optimal solutions, they have been implemented in the DQ algorithm. However, results could not be obtained due to the complex relationships mentioned above. Table 8.9 (next page) provides with the parameter values, for the some of the various test runs that have been conducted and the state of the results obtained from the start iteration to end. It proves the need to understand and research the above mentioned observations to ensure the algorithm provides a solution for optimal problems even if they are badly scaled.

These tests have been run by keeping the optimiser maximum number of iterations to 500 and optimiser tolerances at 10^{-9}. The close to optimal results are optimal when this tolerance is reduced to 10^{-6}. However, this would lead to higher deviations from the non-linear dynamics and are hence not accepted. As can be seen, even for a problem that has an optimal solution, the optimiser faces issues with the scaling parameters. The algorithm has been verified in Chapter 7 and the scenarios with the CQ algorithm. Hence, the need for the implementation of the DQ algorithm requires research on the the relationship between these parameters. Another verification, that needs to be done in future is the use of a different optimiser, to figure out if this is software specific or a problem formulation issue.

With this we conclude this chapter and the next chapter has been dedicated to summarise all the work done till here and present the recommendations for future research on this topic.

Table 8.9: Parameter value summary for test runs with DQ SCvx dynamic algorithm.

Scenario	w_ν	w_η	t_F	State of Results (start -mid - end)
1	100	0.5	6	Close to Optimal - Close to Optimal - Close to Optimal
1	1000	0.5	6	Optimal - Close to Optimal - Close to Optimal
1	1000	1	6	Optimal - Close to Optimal - Close to Optimal
1	10000	1	6	Runs Out of Optimiser Iterations - Numerical Problems - Close to Optimal
1	100	0.5	20	Close to Optimal - Close to Optimal - Close to Optimal
1	1000	0.5	20	Optimal - Close to Optimal - Close to Optimal
1	1000	1	20	Close to Optimal - Close to Optimal - Close to Optimal
1	10000	1	20	Runs Out of Optimiser Iterations - Optimal - Close to Optimal
2	100	0.5	5	Close to Optimal - Close to Optimal - Close to Optimal
2	1000	0.5	5	Close to Optimal - Close to Optimal - Close to Optimal
2	1000	1	5	Close to Optimal - Close to Optimal - Close to Optimal
2	10000	1	5	Close to Optimal - Numerical Problems - Close to Optimal
2	100	0.5	20	Close to Optimal - Close to Optimal - Close to Optimal
2	1000	0.5	20	Close to Optimal - Close to Optimal - Close to Optimal
2	1000	1	20	Close to Optimal - Close to Optimal - Close to Optimal
2	10000	1	20	Runs Out of Optimiser Iterations - Close to Optimal - Close to Optimal

9

Conclusions and Recommendations

This thesis set out to answer the following research question:

How can a spacecraft autonomously map an asteroid for feasible landmarks and descend for a touch-and-go sampling process in a robust and optimal fashion whilst being accurate and safe?

This question has been partially, but successfully answered through this thesis. In the preceding chapters, the successive convex optimal guidance developed by Szmuk et al., 2017 for 6DOF Mars descent has been extended and adapted for asteroid descent using a relative frame of reference and the state represented by dual quaternions. Based on the obtained results a manifold of significant conclusions have been drawn and they will be discussed in this chapter. Also recommendations for future work to further extend this thesis will be discussed.

9.1. Conclusions

A few of the novelties implemented in this thesis work lead us to the conclusions of the thesis. These novelties are:

1. A successive convexification algorithm using DQs has been successfully implemented for a fixed final time asteroid descent problem. The strategy from Szmuk et al. (2017), to convert the non-convex optimal control problem to a sequence of convex optimisation problems, such that a solution to the original problem is achieved, has been modified and adapted. It has been converted from the 6DOF Mars problem in Cartesian coordinates and quaternions to the asteroid descent problem in DQs. At every iteration, the optimisation problem is derived from the non-convex problem and linearised about the solution from the previous iteration. Initialisation is done with a linear trajectory solution to the actual problem and the results show that convergence can be attained even with poor initialisation. The iterative successive convexification method using DQs has been shown to converge in less than 10 iterations within the span of 7 secs for the normalised Mars descent assuming a flat Mars.

2. A modification in the traditional successive convex optimisation method has been implemented, that combines both SCvx method with a dynamic trust region radius to improve the quality of the solution obtained. In every iteration the penalty weight on the trust region is updated depending on the quality of the solution from the previous iteration. The quality of the solution is determined by a comparison of the linear cost obtained and the actual non-linear cost. With this dynamic iterative process, the convergence of the algorithm gives a lower thrust profile, which in turn means a lower propellant consumption as compared to the solution in Szmuk et al. (2017) for the same problem.

   The concern with this method lies in the initialisation of the penalty weights of the trust region and virtual controls. They are lower compared to the Szmuk et al. (2017) by 10^{-2}. The search for these penalty weights is not trivial and has been found by a trial and error method. The dynamic trust region results in a similar lowering of the trust region radius as Szmuk et al. (2017), but lower penalty weight on the virtual controls causes a decrease in them in the range of 10^{-1}, which is comparatively high from the Szmuk et al. (2017)

results. However, the converged solution from the SCvx algorithm has been compared with an RK4 integrator and the difference in the trajectory is in the range of 10^{-3}, which meets the requirements for this thesis.

3. 6DOF relative frame dynamics with DQs has been derived, verified and implemented, which makes the autonomous guidance more adaptive and robust. The previous works in powered descent using successive convexification using Cartesian or model predictive control using DQs are all in the inertial frame of reference, which is a flaw, since most of the constraints need to be in the landing site frame. To avoid this the problem formulation is done in the relative frame, which takes into account the rotation of the asteroid. This enables the SC to always guide itself to the landing site, which has also been parameterised in the relative frame.

   The advantage of the problem formulation in the relative frame could not be utilised in this thesis, due to the problems faced with ECOS regarding finding the correct final time, penalty weights and constraint values. Further research is needed to understand the relation between these three parameters, so that any descent scenario can be implemented in the relative frame of reference using dual quaternions.

4. Attitude constraints for the tilt angle and angular rate of the SC have been successfully implemented using the DQ state by converting them to SOCP constraints. They have been maintained below 10° and 20°/sec respectively. Also the thrust direction constraint using the DQ state, converted to SOCP format has been implemented successfully to keep the gimbal angle within 20°.

   The no-sub-surface-flight, glide-slope and LOS constraints prove to be difficult to handle in the DQ format. This is so, because these constraints are not only quadratic in nature, but also deal with indefinite positive or negative matrices. They also involve Euclidean norms of the quadratic terms and additional terms due to the relative frame dynamics. Although their form has been derived and put in the thesis, further work is required to intelligently break down these constraints or to simply linearise them. A no-subsurface constraint has not been explored, since it would also involve such terms and no-subsurface flight can be obtained just by the glide-slope.

5. Polyhedral gravity field for the asteroid has been introduced in the optimisation problem successfully. The computation cost for the polyhedral gravity field is very high as compared to a flat asteroid assumption or spherical gravity. It is ≈10 secs for every position vector of the SC. This can be reduced by using lower resolution models of the polyhedral asteroid structure and presently the lowest available resolution has been used for Kleopatra. Since, the gravitational field computed from spherical harmonics fall short on accuracy within the Brillouin sphere (for irregular shaped asteroids) and additional models like the Bessel gravity need to be combined with the former for allowable accuracy in the gravity model, these models have not been ventured for this thesis. Also the partial derivative of the polyhedral gravitational field with respect to the dual quaternion in the relative frame has been successfully implemented using complex-step derivate.

6. A ScLERP model for linear interpolation of rotation and translation using dual quaternion transformation has been successfully implemented to generate the first linear solution for the optimisation algorithm.

7. Initial framework and underlying logic from SBMPO has been established for future work. The initial sampling of the obstacle free configuration space along with the control action space has been implemented. A minimum volume ellipsoid to enclose the asteroid and build the obstacle space has been used. An illumination model of the asteroid depending on the Sun position has been implemented and updated with the asteroid rotation. The initialisation of trajectories using random control actions bounded by maximum ΔV and propagation until collision has been achieved.

In conclusion, a promising algorithm for descent on an asteroid and initialisation of a state-of-the-art mapping algorithm has been implemented. However, further development is required to make it applicable to an actual test case. This leads us to the next steps to be taken in this research and they have been presented in the recommendations.

9.2. Recommendations

This thesis just lays down the foundations of guidance algorithms and there are still many challenges, that need to be overcome to make it a complete method. Some of the most important recommendations for immediate future research work are given below:

- The SMBPO algorithm needs to be completed incorporating the heuristics for Δv update. After the initialisation, the stages of trajectory scoring for viewing capabilities and fuel consumption need to be implemented to build the optimisation algorithm. Search heuristics need to be implemented to bias more trajectories towards interesting sites on the asteroid, therefore, refining the configuration space with every iteration of the algorithm. Ultimately, this should lead to lower global mapping phase times for actual missions, since automated trajectories would lead to the spacecraft spending more time on well illuminated and interesting sites to select a descent site for the TAG descent phase.

- Several enhancements are needed for the SCvx descent algorithm to be complete. Firstly, the glide-slope and line of sight constraint . Secondly, an optimal time to go. Thirdly, higher order discretisation has been encoded in the algorithm, but needs some work to be actually implemented. This needs to be considered to compare with the accuracy of the obtained results from first order discretisation. Fourthly, we implement ZOH control, since the propellant optimisation is a bang-bang optimisation problem. However, better control can be obtained with FOH and should be looked into as further work.

- A deeper insight to the formulation of the SCvx problem needs to be obtained. The exact relation between the scale of the penalty weights for virtual controls and trust region and non linear constraints has to be figured out. At this point, the only finding is that the penalty weight values, the propellant mass to be optimised and the sensitivity of ECOS to scaling are related. Further tweaking of these values is required to figure out the exact relation. This will help bound the values of allowable penalty weights for any powered descent problem making the algorithm easier to comprehend.

- Another effort to make the SCvx algorithm more global is to work with different landing scenarios. In this thesis the scenarios have been built to avail a solution, prove the algorithm is functioning and that it provides optimal results. With the above two points in place and additional landing scenarios, the algorithm can be made global for all powered descent scenarios in atmosphereless environments.

- To make the SCvx algorithm compatible with a GNC system, a closed loop GNC simulator needs to be built. With the work from Razgus (2017) and this thesis and a control algorithm in place, the overall quality of working with DQs can be evaluated. HDA, should also be incorporated within the loop to verify the capabilities of the SCvx algorithm to deal with diversions. SRP and 3^{rd}BP perturbations should also be added to obtain a more accurate achievable precision.

- To evaluate the advantages of the SCvx guidance, it is necessary to have comparisons with other guidance technologies available. For example, quadratic programming with model predictive control using DQs has already been explored by Lee and Mesbahi (2017) and a comparison with their results could provide a better picture of whether the SCvx optimization method is more capable and provides better precision for powered descent. Since, theoretically convex optimisation provides with a global optimal propellant usage, these comparisons would help confirm or deny it.

- For this thesis, only flat asteroid or polyhedral gravity field have been assumed for the optimisation problems. Simulations with spherical gravity and gravity from spherical harmonics should also be considered in future work. Gravitational field computation by spherical harmonics for irregularly shaped asteroids as used by Pinson and Lu (2016) proves that a combined spherical harmonics model outside the Brillouin sphere and a Bessel model inside the same provides the best computational efficiency. The error in the computed gravity ranges from 5 to 10 % for nearly spherical and elongated asteroid shapes respectively. For a compromise over computational cost, these models should be incorporated and the margin of error in the landing site can be computed in turn depending on the error in the computed gravity.

- ECOS has been proved to have a sensitivity to the scale of the problem parameters and hence, use of other convex optimisers with the SCvx problem needs to be investigated for better performance under badly scaled problem parameters. SEDUMI, SDTP3, MOSEK, GUROBI are some open source or free academic license solvers, that can be used to approach the SCvx method and check for a better performance.

- The optimisation problems for both mapping and descent using SMBPO and SCVx should be coded using compiled languages like C or C++ to obtain better computational speeds. This thesis has already proved the polyhedral gravity fields result in higher computational time and this can be reduced with compiled languages substantially. For real time applications, this becomes a necessity rather than a choice and hence even if the complications for coding in the compiled languages are higher, it should be looked into.

A

Global Mapping Guidance using Sampling Based Motion Planning

In the mapping phase, the guidance subsystem has to generate a trajectory or number of trajectories for the SC to follow in order to observe the entire target body. It has to take care of the attitude of the SC such that the target body is never out of sight and within a certain constrained viewing angle for generating an accurate map. Also maintaining the desired altitude is necessary whilst orbiting or performing orbital manoeuvres around the asteroid to generate high-resolution maps.

Utashima (1997) in his study has presented with SC orbits possible around asteroids in order to observe them globally with high resolution. There are two types of orbits possible for mapping the asteroid, frozen and polar orbits. Frozen orbits are those whose Keplerian mean elements do not change given that solar radiation pressure is the only disturbance. Two kinds of frozen orbits are possible, one in the orbital plane of the asteroid and the other called the solar plane-of-sky orbit is an orbit that always faces the Sun. The prior one is not adequate for high-resolution observation whereas the latter can be adequate for very small bodies with radii below 2-3 km. In case the altitudes of these orbits are lowered, they are highly sensitive to the oblateness of the asteroid, and the frozen orbits need to be maintained by manoeuvres costing $\approx$ 100m/s for a year's worth of maintenance.

Polar orbits, on the other hand, are best suited for global mapping which negates the effect of the asteroid's oblateness. Also, the only maintenance required is to make the eccentricity zero. It has also been proved if the ratio of characteristic angular velocity due to the radiation pressure to that from oblateness multiplied by the limit for eccentricity is less than 1, maintenance is not required at all for polar orbits.

In the recent mission of MUSES-C to Ryugu, the SC hovered at a distance of 20 km from the asteroid for $\approx$ 20 days to completely map it.[1] In the OSIRIS-Rex mission to Bennu, the detailed survey phase where mapping of the asteroid is planned, the SC conducts hyperbolic flybys with the closest approach of 3.5 km over four specific locations(GalEdd and Cheuvront (2015)). Images and laser altimetry from each of these locations for a complete asteroid rotation are conducted. These provide with a resolution of the surface features up to $\approx$ 21 cm and accuracy of the map within $\approx$ 5 m (Beshore et al. (2015)). Then it conducts 5 km flybys to study its geometry for details of the surface properties from locations above the equator with independent illumination angles (due to multiple local times of day on the surface). By the end of this phase, 12 sampled sites are identified for sample collection.

The mission design of OSIRIS-Rex provides more autonomy to the SC, to map and select a landing site and in this thesis, we will focus on the same. As discussed in Chapter 2 we would approach this problem with *sampling based motion planning optimisation* to reduce the propellant cost and time cost to reduce the period of topographic mapping whilst being within a limited Δv utilisation. We will approach this problem using dual quaternion relative states which have not been done before.

The IVP problem for mapping is found on the next page, and it will be formulated as per the requirements of sampling based motion planning in the further sections. Due to constraints of time, the complete implementation has not been possible, but the basic ideology of the method has been researched, and initialisation has been achieved. This can be used in future works to make it a complete guidance algorithm.

[1] http://global.jaxa.jp/projects/sat/hayabusa2/topics.html#topics12394, date accessed:04-03-18

➽ PROBLEM 2A: Mapping phase general guidance problem

Given: Initial states at time $t = t_0$

$$\boldsymbol{r}(t_0) = \boldsymbol{r}_0, \quad \boldsymbol{v}(t_0) = \boldsymbol{v}_0, \quad m(t_0) = m_0 \tag{A.1}$$

where $\boldsymbol{r}(t)$ is the position, $\boldsymbol{v}(t)$ the velocity and $m(t)$ the mass of the SC at time t. Find an acceleration profile and thus a trajectory

$$\boldsymbol{a}(t) = \frac{\boldsymbol{T}(t)}{m(t)} + \boldsymbol{g}(t), \quad \forall t \in [t_0, t_{end}] \tag{A.2}$$

where $\boldsymbol{a}(t)$ is the acceleration, $\boldsymbol{T}(t)$ is the thrust, $m(t)$ is the instantaneous spacecraft mass, and t_end is an open final time, such that the spacecraft reaches an open final state

$$\boldsymbol{r}(t_{end}) = \boldsymbol{r}_{end}, \quad \boldsymbol{v}(t_{end}) = \boldsymbol{v}_{end} \tag{A.3}$$

after satisfying mission objectives of observation

$$MO_{completion} = 100\% \tag{A.4}$$

while satisfying some given constraints

$$\begin{aligned} \boldsymbol{g} &\leq 0 \\ \boldsymbol{h} &= 0 \end{aligned} \tag{A.5}$$

This problem is usually solved analytically during mission planning and the mission profile is already onboard the SC to follow through directly. In this thesis we will extend on the novel method -*sampling based motion planning*- to enable the SC to solve this problem autonomously in real time.

A.1. Basic Mapping GNC framework

The basic architecture for global mapping is presented in Figure A.1. As shown in the flow, the primary navigation with the help of onboard instruments (Initial Survey phase) provides with better maps of illumination, gravity, rotation, mass, density and size of the asteroid. This is then fed to generate a risk map to prevent the escape of the SC from the asteroid gravitational field helping constrain the propellant expulsion actuator from exceeding thrust values. This is fed then to a simplified guidance logic that can be used to generate cost maps of fuel and observation. Once these initial maps are obtained, they can be fed to the actual guidance algorithm to calculate trajectories and provide the required control inputs to the actuators. The SC then follows the trajectories, with errors in its state from limitations of modelling various perturbations and actual actuator control outputs. These are estimated by the navigation system and fed back to the MVM to decide

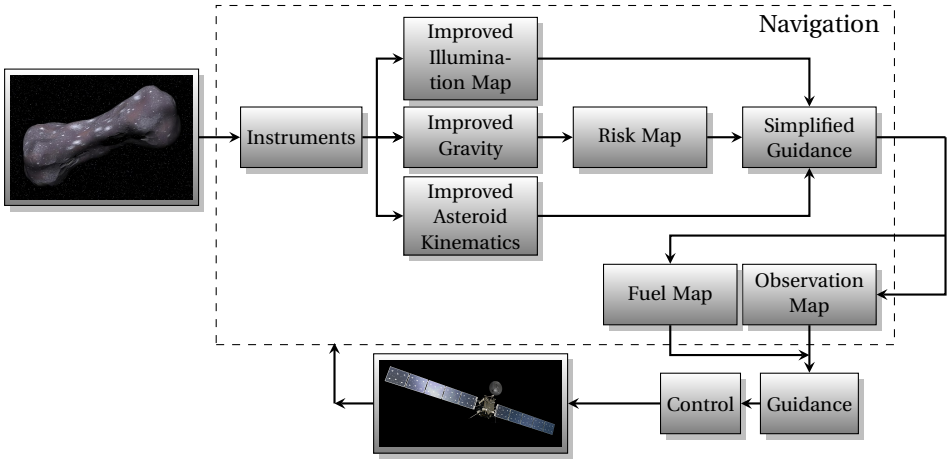


Figure A.1: Top level GNC architecture for global mapping of the small body

on whether new trajectories need to be calculated.

A.2. Refined Problem Formulation

With the mission scenario established, the mapping problem is reformulated. The guidance problem for mapping can be defined as:

The guidance problem for global mapping can be defined as finding a number of fuel optimal trajectories from a given initial state (pose and attitude) to observe the entire surface of the asteroid ensuring satisfactory illumination of the patch under observation and avoiding any collision with the asteroid, whilst being under a variable gravity field and acted on by disturbances due to solar radiation pressure and 3rd body perturbation.

➽ PROBLEM 2B: Refined mapping phase 6-DOF guidance optimisation problem

minimise $f(\Delta v^{\mathtt{I}}_{\mathtt{B/I}}(t)) + f(r^{\mathtt{I}}(t), t_{obs})$

subject to:

Kinematics:
$\dot{\boldsymbol{r}}^{\mathtt{I}}(t) = \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}}(t), \qquad \dot{\boldsymbol{q}}_{\mathtt{B/I}}(t) = \frac{1}{2}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \otimes \boldsymbol{q}_{\mathtt{B/I}}(t)$

Dynamics:
$\dot{\boldsymbol{v}}^{\mathtt{I}}_{\mathtt{B/I}}(t) = \frac{\boldsymbol{T}^{\mathtt{I}}(t)}{m(t)} + \boldsymbol{g}_{\mathtt{I}}$
$\dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/I}}(t) = \mathbf{J}^{-1}(\boldsymbol{r}_{\mathtt{B}}(t) \times \boldsymbol{T}^{\mathtt{B}}(t) - \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \times \mathbf{J}\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t))$

Control Constraints:
$\|\Delta \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}}(t)\| \leq \Delta v_{max}$

Boundary Conditions:
$\boldsymbol{r}^{\mathtt{I}}(0) = \boldsymbol{r}^{\mathtt{I}}_0, \quad \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}}(0) = \boldsymbol{v}^{\mathtt{I}}_{\mathtt{B/I}_0}$
$\boldsymbol{q}_{\mathtt{B/I}}(0) = \boldsymbol{q}_{\mathtt{B/I}_0}, \quad \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(0) = \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}_0}$

Additional Constraints:
$M_c = f(n, t_{obs})$

where, t_{obs} is the time for which a particular patch is observed, n is the number of patches to be observed and M_c is the mission completion status.

The concept of Motion Planning has been developed for robotics and artificial intelligence planning. It is as the name means, planning the motion of an object. In a simple scenario, it can be defined as figuring out a trajectory, that links two static positions while avoiding obstacles. The problem can, therefore, be charted via discretisation and a solution can be found using graph search methods like A^* or Rapidly exploring Random Trees (RRT). Motion planning in the last few decades has found applications in the autonomous ground and air vehicles. The main difference in these applications is the involve spatiotemporally consistent system dynamics as compared to the robotics, that deals with just the kinematics of the system. It was developed likewise, on the lines of control theory having feedback policies enabling adaptive response during execution and a focus on stability, to ensure that the system dynamics do not drive it out of control.

In recent works, an emphasis on optimisation of resource consumption like energy and time has been established. Also, uncertainties in the system model or control system can be taken into consideration. Trajectory smoothing can also be applied at the end to satisfy all the constraints of the complete continuous system. It can be seen, that with these factors the complexity of even the simple scenario put forth previously is substantially increased. For the mission of asteroid mapping, the trajectory should satisfy observational requirements, whilst satisfying a minimum fuel consumption.

Surovik and Scheeres (2014) and Capolupo et al. (2017) have established algorithms for mapping and descent to asteroids using RRT and sampling-based motion planning respectively. These are used to develop the base algorithm and then convert it to an algorithm based on DQs.

A.3. Basic Concepts of Motion Planning

Motion planning is a development of the 18^{th} century. The use of motion planning algorithms in robotics and automation has been well established in the past decade. Their introduction to space applications is very recent, but have proved to solve optimisation problems involving high complexities with ease. This section will go through some basic concepts of motion planning to help with the more detailed sections for the working of the algorithm. The algorithm can be defined in the simplest way as,

Input The initial state of the SC, desired goal objectives, the geometric description of the SC and obstacle region and constraints to the motion of the SC.

Output The precise description of how the SC moves from its initial state to the next state achieving the goal objectives, satisfying the constraints on its motion and avoiding the obstacle region always.

A.3.1. Geometric Representations

Motion planning problems require the geometric models of a system of bodies in a space to be defined so they can be manipulated while solving the problem. The two general ways of defining a complicated geometric model are *boundary* or *solid* representation. For a 3D problem, we need to first define a *world*, $\mathcal{W} \in \mathbb{R}^3$. This world pertains to the two entities of interest,

Obstacles, $\mathcal{O}$:
These are the spaces in the world, that are permanently occupied, in our case the target small body and the obstacle space being time dependent due to its rotation.

Robots, $\mathcal{A}$:
These are the spaces, geometrically modelled and controllable through the motion plan, in our case a simplified SC model.

A solid representation for $\mathcal{O}$ needs to be developed for the target asteroid, which is developed in terms of a combination of *primitives*. For a complicated obstacle like the small body, a finite number of Boolean combinations is the way to go. A finite number of unions, intersections and set differences of primitives can be used to define $\mathcal{O}$ by set theory. Also, all these primitives are subsets of the world.

Polyhedral models

From the definition of convex sets in the Section 6.1.1, it can be seen, that defining a space using convex polyhedra has the advantage, of every point lying on the line connecting any two points in each polyhedron being in the set of the polyhedron. This means, that it becomes a closed space with no pockets or indentations. Each vertex of, say an m-sided polyhedron, the line segments joining the two vertices and the polygon embedded between a triplet of vertices called the face become the three primary features of the polyhedron. Each primitive is defined by a half-plane, H_i for $i \leq\leq m$, as a subset of $\mathcal{W}$, which can be represented as below,

$$H_i = \{(x, y, z) \in \mathcal{W} \mid f_i(x, y, z) \leq 0\} \tag{A.6}$$

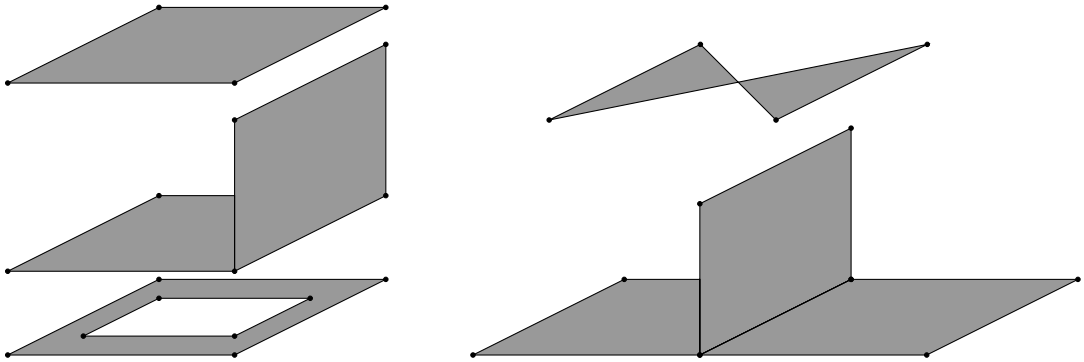
which describes all the points on one side of the line $f_i(x, y, z) = 0$ and the points on that line. The convex, polygonal obstacle space can, therefore, be defined as,

$$\mathcal{O} = H_i \cap H_{i+1} \cap \ldots \cap H_m \tag{A.7}$$

The complexity is in the shape of the asteroid, since it can have concave features, which make the entire obstacle space non-convex. Hence the intersection of half-planes falls short of representing the non-convex polyhedral structure of the obstacle space. For our test case, $\mathcal{O}$ is a non-convex, polygonal subset of $\mathcal{W}$ represented by the union of convex, polygonal sets $\mathcal{O}_i$,

$$\mathcal{O} = \mathcal{O}_i \cup \mathcal{O}_{i+1} \cup \ldots \cup \mathcal{O}_m \tag{A.8}$$

The decomposition of this nonconvex subset is not unique and can be decomposed, such that computational performance is optimised. This is, however, out of scope for this thesis and we use the already available polyhedral files developed by NASA for observed asteroids. They are built on triangular adjoint facets, defined by edges and normals to obtain the actual shape of the asteroid. The models can be found in Chapter 7.



(a) Allowable manifolds, since they represent a $\mathbb{R}^2$ configuration space.

(b) Non-allowable manifolds, since they are not continuous at the points where branching occurs.

Figure A.2: 2D manifolds representing configuration space.

A.3.2. Configuration Space

Motion planning problems are described in the entire region of space called the world, denoted by $\mathscr{W}$ and for a 3D world $\mathscr{W} \subset \mathbb{R}^3$. The SC and asteroid are analogous to the motion planning definitions of robot and obstacle region respectively and are nothing but polyhedral structures in 3D space. The *obstacle space*, $\mathscr{O}$ is a subset of the world $\mathscr{O} \subset \mathscr{W}$. The SC, however, is defined in a different space called the *configuration space* or $\mathscr{C}$-space. It is the set, that contains all the rigid transformations, that the SC can undergo. The motion planning algorithm basically computes a path in the part of $\mathscr{C}$-space, that avoids the obstacle space.

The $\mathscr{C}$-space in the context of control theory is a *differential manifold*, but since motion planning $\mathscr{C}$-space does not involve calculus, it is simpler to define it as a *topological manifold*. Figure A.2a and A.2b represent 2D allowable topological manifolds and ones that are not allowed respectively. An n-dimensional topological manifold is a subset of $\mathbb{R}^m$, where $n \leq m$, such that every state, which is an element of $\mathscr{C}$ is contained in at least one open subset of $\mathscr{C}$, that is homeomorphic[2] to $\mathbb{R}^n$ (LaValle (2011)). For 3D dimensional space, the rotational motion can be defined using quaternions and visualised as a sphere of axes, about which rotation can take place and a circle about each axis for the amount of rotation, technically called a *real projective 3-space* and denoted by $\mathbb{RP}^3$. The 3D $\mathscr{C}$-space using quaternions to represent rotational motion and Cartesian coordinates for translation the $\mathscr{C}$-space is, therefore, represented as $(x, y, z, q_1, q_2, q_3, q_4)$ and $\mathscr{C} \subset \mathbb{R}^3 \times \mathbb{RP}^3$. Extending this to dual quaternions, the $\mathscr{C}$-space is represented as $(q_{r_1}, q_{r_2}, q_{r_3}, q_{r_4}, q_{d_1}, q_{d_2}, q_{d_3}, q_{d_4},)$ and $\mathscr{C} \subset \mathbb{RP}^3 \times \mathbb{R}^3$.

With the $\mathscr{C}$-space defined for the 6DOF problem, a definition for the space preventing collision also needs to be provided. These regions are prohibited for the SC to enter. Let the SC transformed to a configuration p be defined by $\mathscr{A}(p) \subset \mathscr{W}$, which is a closed set of points. Now this configuration p is said to maintain the state of no collision iff $\mathscr{A}(p) \cap \mathscr{O} \neq 0$, which means that no configuration of $\mathscr{A}$ lies in the obstacle space, $\mathscr{O}$. This space is called *free space* and is defined as,

$$\mathscr{C}_{free} = \{x \in \mathscr{C} | \mathscr{A}(x) \cap \mathscr{O} = 0\} \tag{A.9}$$

The complement of this space is nothing but the obstacle space itself, $\mathscr{O} = \mathscr{C}_{obs} = \mathscr{C} \backslash \mathscr{C}_{free}$. With this the problem can be mathematically defined as (LaValle (2011)):

Given an SC description, $\mathscr{A}$, and asteroid description, $\mathscr{O}$, a configuration space, $\mathscr{C}$, an initial configuration, $x_{int} \in \mathscr{C}$ and a goal configuration, $x_{end} \in \mathscr{C}$, compute a free continuous path, $\tau \rightarrow \mathscr{C}_{free}$ with $\tau_0 = x_{int}$.

There are eight underlying assumptions for this statement, which can be found in Ghallab et al. (2004) for reference. These assumptions make the motion planning a *restricted model* in terms of a deterministic, static, finite and fully observable *state transition system* with restricted goals and implicit time (Ghallab et al. (2004)). These assumptions can be relaxed to cater to the application or problem at hand. With the basic concepts of motion planning, we can go ahead with the step by step formulation of the mapping problem.

[2] If for an open set, $\mathscr{O}$, there exists a continuous, bijective function $f : \mathscr{O} \rightarrow \mathbb{R}^n$, for which the inverse f^{-1} is also continuous, $\mathscr{O}$ is homeomorphic to $\mathbb{R}^n$

A.4. Sampling Based Planning

The basic logic of sampling-based motion planning is to avoid explicitly constructing the $\mathscr{C}_{obs}$ and probe into $\mathscr{C}$ by means of a sampling scheme. This can be accomplished by means of the motion planning algorithm, that detects a collision. This allows the development of an algorithm, that is independent of the geometric representations of $\mathscr{C}_{obs}$ or $\mathscr{A}$. Figure A.3 shows the flowchart of sampling based planning, where the collision detection block is the *blackbox* of the motion planning system. In the subsections, we will discuss the functioning of the collision detection and the sampling based algorithm.

A.4.1. Discrete Searching/Planning

Discrete searching or planning is the initial part of the planning algorithm, that determines the sequence of actions to achieve a certain goal. Since the application is already complicated, all forms of uncertainty are avoided to avoid involving probability theory. Therefore, the models will be known through and through and will be predictable. Hence these models are also called *predictive models*.

As required by any kind of problem formulation to define a discrete feasible plan, a model of the state space is required, which is nothing but a set of all possible situations, that $\mathscr{A}$ can be in and discussed as configuration space, $\mathscr{C}$ earlier. The planner then chooses *actions*, $\boldsymbol{u}_k$ at sequence step k to transform the *current state*, $\boldsymbol{x}_k$ to the *new state*, $\boldsymbol{x}_{k+1}$. This can be represented by a *state transition function*.

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{A.10}$$

A new space called *control* or *action* space, $\mathscr{U}$ is defined for states at every sequence step. $\mathscr{U}_k$ and $\mathscr{U}_{k+1}$ need not be disjoint and could hold the same actions. For all possible actions for the states of all sequence steps, the space is defined as,

$$\mathscr{U} = \bigcup_{\boldsymbol{x} \in \mathscr{C}} \mathscr{U}_k \tag{A.11}$$

Additional to the action space in the search phase a set of goal states, $\boldsymbol{x}_{end}$ is defined, such that $\mathscr{C}_{end} \subset \mathscr{C}$. The planning algorithm has to now equip itself to find the path, τ, which is a sequence of actions, that transforms the state from $\boldsymbol{x}_{int}$ to $\boldsymbol{x}_{end}$. An easy way to visualise the planning is to build a *state transition graph*, with vertices as every state from the finite $\mathscr{C}_{free}$. A *directed edge* from $\boldsymbol{x}_k \in \mathscr{C}_{free}$ to $\boldsymbol{x}_{k+1} \in \mathscr{C}_{free}$ exists iff and action $\boldsymbol{u}_k \in \mathscr{U}$ exists, such that Eq. (A.10) is satisfied. Figure A.4 represents a simple example of a state transition graph for moving from an initial state to final state on an infinite tile floor while avoiding $\mathscr{C}_{obs}$.

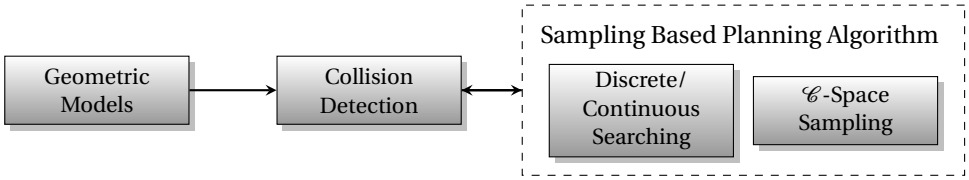


Figure A.3: Sampling based planning flowchart (LaValle (2006)).

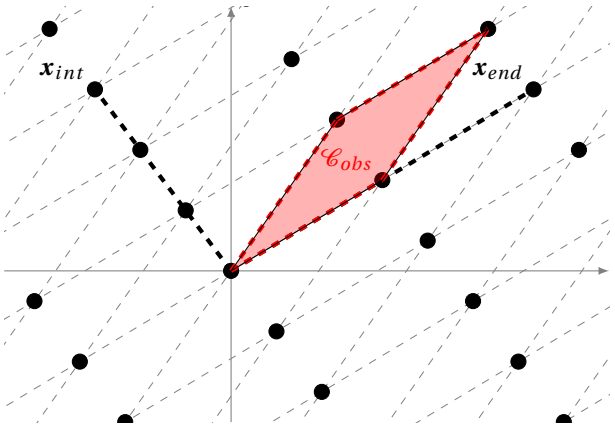


Figure A.4: State transition graph for moving from an initial state to final state on an infinite tile floor.

Search Schemes
There are various methods to search for the trajectories for example forward search, backward search or bi-directional search. They work from the initial state to the end or goal state, vice versa or work towards each other respectively. Each of these methods builds a search graph, $\mathcal{G}$ incrementally as it explores more states in the state transition graph. Each search graph, $\mathcal{G}$ consists of the explored vertices, $\mathcal{V}$ and their directed edges, $\mathcal{E}$. More complicated search schemes can be developed, but for a basic ideology, these are put forth. These schemes can be extended to a continuous state space, and they are summarised below for an overview.

Initialisation
Initially $\mathcal{G}(\mathcal{V},\mathcal{E})$ has an empty set of $\mathcal{E}$ and $\mathcal{V} = \{\boldsymbol{x}_{int}\}$ or $\{\boldsymbol{x}_{end}\}$ or $\{\boldsymbol{x}_{int}, \boldsymbol{x}_{end}\}$ depending on forward, backward or bi-directional scheme. The generation or revelation of more states in increments is done by the *trees* growing from one or two states to reveal more of the state transition graph in increments. In case the scheme asks for more than two trees extending initially, $\mathcal{V}$ contains more states.

Vertex Selection and Action Implementation
Next step is to expand the currently selected state, $\boldsymbol{x}_{cur} \in \mathcal{V}$, by means of a priority queue, that indexes the next states. An action, $\boldsymbol{u} \in \mathcal{U}_{cur}$ is applied to the current state to obtain a new state, $\boldsymbol{x}_{new}$.

Directed Edge Insertion
Additional requirements besides satisfying Eq. (A.10) can be set for a path/edge to be valid. These are specific to the application, and pass-fail criteria for them can be used to determine if the particular edge from $\boldsymbol{x}_{cur}$ to $\boldsymbol{x}_{new}$ should be generated or not. Once a valid edge can be found and $\boldsymbol{x}_{new}$ is not already in $\mathcal{V}$, it is inserted as an explored state and $\mathcal{G}(\mathcal{V},\mathcal{E})$ is updated with the new edge and vertex.

Solution Check
If $\mathcal{G}(\mathcal{V},\mathcal{E})$ has already encountered the edge from $\boldsymbol{x}_{int}$ to $\boldsymbol{x}_{end}$ and there exists only one such single tree, that is of course the solution. The existence of just a single tree from $\boldsymbol{x}_{int}$ to $\boldsymbol{x}_{end}$ obviously makes the algorithm trivial, but that is usually not the case, and more possibility of such connected trees makes the planning expensive. This also calls for an optimisation process to find the best possible tree for the solution.

Iterate
Iterations have to continue till the solution tree is obtained and all actions have been implemented. The algorithm should include additional constraints to terminate the tree in case the constraints are violated and report failure in case that happens.

Logic Based Planning Scheme
Although the above schemes form the basis of motion planning algorithms, they fall short in applications with enormous state spaces for example if $n(\mathcal{C}) = 10^{100}$ as is also the case of our asteroid mapping problem. Here a logic-based planning scheme needs to be implemented, such that the entire state space need not be searched or explored. Additionally, if the application involves some sort of regularity for example periodicity of orbits, a more compact representation for the same can be used with a logical scheme.

Another benefit, is logic-based representations provide solutions, that can be traced back to the steps involved to arrive at the goal. The only disadvantage is that these schemes cannot be generalised over applications and have to be customised based on whether the state space is continuous and if there is unpredictability involved. Also, multiple decision makers and sensing uncertainty have to be considered for logic-based schemes. This calls for particular application specific heuristics, to redefine the sampling. Here we briefly discuss the main approach of *building a planning graph* and *planning as satisfiability* schemes (can be found in detail in LaValle (2006)).

A *planning graph* entails a way to utilise a data structure for information about *reachable* states. The logic is to construct a smaller graph than the state transition graph, which provides information about states, that can be possibly reached, eliminating all the states, that are impossible to reach. This helps the algorithm perform a refined search rather than checking all states in the state transition graph considerably reducing the computation cost. The planning graph, therefore, helps in the development of search heuristics for the problem at hand.

Planning as satisfiability can be used to convert the entire problem to Boolean constructing each partial plan or the complete plan to give an outcome in terms of a "TRUE" or "FALSE" value. However, it can also be used in conjunction with the *planning graph* scheme where certain partial plans can generate Boolean outcomes. In Surovik (2016) this conjunctive approach has been used and will be explored in this thesis. In Section A.5 the scheme and heuristics required for the mapping phase are discussed in detail.

A.4.2. Sampling

The concept of sampling is, as the name suggests, visualising the state space for motion planning in terms of samples. This is, because a continuous state space is countably infinite, but to find a solution in limited time, we need to examine a finite numbered sample space. The logic of sampling is, therefore, to represent the state space, in such a way that convergence to a solution can be achieved. As can be said intuitively, the performance of the algorithm is crucially dependent on the sampling technique. After careful selection of a sampling technique based on the application, it is to be used for; discrete planning can be used. Two concepts need to be discussed for sampling, firstly *metric spaces* and secondly *measure spaces*.

In most planning problems, the sampling of $\mathscr{C}$ results in a function, that determines the distances between the points in $\mathscr{C}$. This results in a *metric space*, $(\mathscr{X}, \rho)$, where $\mathscr{X}$ represents a topological space and ρ a function, such that $\rho : \mathscr{X} \times \mathscr{X} \rightarrow \mathbb{R},\ \forall a, b, c \in \mathscr{X}$. The function ρ should have the properties of non-negativity, reflexivity, symmetry and triangle inequality, which can be found in detail in LaValle (2006). Since DQs, represent both the position and orientation of the SC ($\mathscr{A}$), *SO(3)* metric, which can be used for quaternions is not enough. *SE(3)* metrics are required for DQs, to represent the set of all possible rigid-body transformations in 3D space.

$$SE(3) = \left\{ T \in \mathbb{R}^{4\times4} \,\middle|\, T = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{x} \\ \boldsymbol{0}_{3\times1} & 1 \end{bmatrix},\ \boldsymbol{C} \in SO(3), \boldsymbol{x} \in \mathbb{R}^3 \right\} \tag{A.12}$$

where, $\boldsymbol{x}$ defines the position vector of $\mathscr{A}$ and $\boldsymbol{C}$ is the rotation parameterised in *SO(3)* by the unit quaternion in the DQ. In case the sampling is done in the space of dual vectors, the metric space is simply an Euclidean metric for each half of the dual vector.

Sampling planning often requires defining volumes in the metric space. A *measure space* or *measure* is a function, that produces a subset of the metric space. Using *sigma algebra* (properties can be found in LaValle (2006)), $\mathscr{X}$ is divided into a collection of subsets $\mathscr{B}$, such that the union and intersections of these subsets also lies in $\mathscr{B}$. These sets are called *measurable sets*, defined as a simple set, B of open balls in $\mathscr{X}$ as below,

$$B(\boldsymbol{x}, r) = \{\boldsymbol{x}' \in \mathscr{X} \mid \rho(\boldsymbol{x}, \boldsymbol{x}') \leq r\} \tag{A.13}$$

The *measure* function is now defined as a function $\mu : \mathscr{B} \rightarrow [0, \infty]$, that satisfies the axioms (LaValle (2006)),

1. If a set B is empty, $\mu(\emptyset) = 0$.
2. For a collection of measurable sets, B_i, the measure must satisfy $\mu(B) = \sum_i \mu(B_i)$, where B represents the union of the collection of measurable sets.

A.4.3. Collision Detection

Once the sampling is done, the next step is to make sure $\mathscr{C}$ does not intersect with $\mathscr{C}_{obs}$. This is referred as collision detection and this is usually treated as a blackbox, but since it adds to the computational cost, it must be taken into account. There are various methods to detect a collision, but for this thesis *checking the path segment* will suffice. This is basically checking any generated path, $\tau : [\boldsymbol{x}_{ini}, \boldsymbol{x}_{end}] \rightarrow \mathscr{C}$ to be $\tau([\boldsymbol{x}_{ini}, \boldsymbol{x}_{end}]) \subset \mathscr{C}_{free}$. Every node in the path is checked for this criteria and a resolution of the path is fixed to avoid missing nodes, that could be dissatisfying the same.

A.4.4. Need for Optimisation

A need for motion planning is the notion of *completeness* (LaValle (2006)). What this means is that the planning algorithm has to guarantee a solution to the problem at hand. If the algorithm is capable of reporting the availability or unavailability of a solution within a finite amount of time it is considered to be *complete*. Such completeness cannot be achieved by sampling based planning. Most sampling techniques accommodate *probabilistic* completeness. This is, because highly dense sampling can ensure that the algorithm converges to a solution if it exists, however, the algorithm could run forever in case there is not a solution. Hence we need to move to *combinatorial* motion planning, that involves sampling based methods, along with control theory and optimisation. The next section is dedicated towards the combinatorial planning for small body observation as prescribed in Surovik (2016), made simpler for adapting to a DQ implementation.

A.5. Sampling Based Model Predictive Optimisation

A.5.1. Predictive Model

The predictive model holds the controllability of the algorithm, by defining, the motion of $\mathscr{A}$ (which is the SC in our case), the high-level mission objectives and the quality of the trajectory based on the objectives that

have been met.

Physical System

This basically models the dynamics of the system and the reference frames, which have been discussed in Chapter 4. It also holds the polyhedron model of the $\mathscr{C}_{obs}$, in our case an asteroid. It contains the polyhedral gravity field of the asteroid and also the disturbance models of 3$^{\text{rd}}$BP and SRP. Reiterating them,

Relative dynamics and kinematics

$$\dot{\check{\boldsymbol{q}}}_{\mathtt{B/A}} = \frac{1}{2}\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} \check{\otimes} \check{\boldsymbol{q}}_{\mathtt{B/A}}$$

$$\check{\boldsymbol{F}}^{\mathtt{B}} + \check{\boldsymbol{F}}^{\mathtt{B}}_{D} = \check{\boldsymbol{J}}\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/A}} + (\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}) \check{\times} \check{\boldsymbol{J}}(\check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}}) + \check{\boldsymbol{J}}\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{A/I}} \check{\times} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/A}} + \check{\boldsymbol{J}}\dot{\check{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{A/I}} \check{\times} \check{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{A/I}} \check{\times} \boldsymbol{R}$$

Polyhedral gravity

$$\boldsymbol{g}^{\mathtt{A}} = \nabla U = -G\varrho \sum_{e \in edges} \boldsymbol{E}_e \, \boldsymbol{r}_e \, L_e + G\varrho \sum_{f \in faces} \boldsymbol{F}_f \, \boldsymbol{r}_f \, \omega_f$$

Disturbances

$$\boldsymbol{T}^{\mathtt{B}}_{GG} = \sum_{i=1}^{n} \boldsymbol{\rho}_i \times m_i \, \boldsymbol{g}^{\mathtt{B}}_i$$

$$\boldsymbol{a}^{\mathtt{B}}_{3BP} = -G\frac{m_d}{r_d^2}\sqrt{1 + \frac{1}{(1-2\gamma\cos\alpha+\gamma^2)^2} - \frac{2(1-\gamma\cos\alpha)}{(1-2\gamma\cos\alpha+\gamma^2)^{3/2}}}$$

$$\boldsymbol{T}^{\mathtt{B}}_{3BP} = \boldsymbol{r}^{\mathtt{B}}_i \times m_i \, \boldsymbol{a}^{\mathtt{B}}_{3BP}$$

$$\boldsymbol{F}^{\mathtt{B}}_{SRP} = \sum_{i=1}^{N} \boldsymbol{F}^i_{SRP} = \sum_{i=1}^{N} -P_{\odot} A_i \cos\theta_i \left(2\left(\frac{\alpha^i_d}{3} + \alpha^i_r \cos\theta_i\right)\boldsymbol{n}_i + (1-\alpha^i_r)\boldsymbol{e}_{\odot,i}\right)$$

$$\boldsymbol{T}^{\mathtt{B}}_{SRP} = \sum_{i=1}^{N} \boldsymbol{r}_i \times \boldsymbol{F}^i_{SRP}$$

Mission Objectives and Outcomes

In this stage of the predictive model, the physical motion of the SC through space in terms of mission outcomes needs to be established. This is, so that we can ensure safety of the SC and its ability to observe the asteroid as required and to also be certain about the practicality of the plan. The mission outcomes could be Boolean (true or false) or real valued outcomes, that help evaluate the trajectories in terms of objective fulfilment.

Safety Constraints

The first safety constraint for the mapping phase is to prevent the impact of the SC on the target body. This is checked by Boolean failure criteria, that says zero (false) in case of no impact and one (true) otherwise. In case of asteroids due to their irregular shape, just the computation of the second order differential of the gravity potential ($\nabla^2 U(r^{\mathtt{A}})$) or the solid angle ($\omega_f$) as stated in Eq. 4.109 can be used. Another way to avoid the computation time of ω_f for each new position of the SC, a minimum volume ellipsoid can be generated based on the maximum asteroid dimensions along the X, Y and Z axes. Moreover, a violation of the volume, i.e., the SC being inside this ellipsoid has a much lower computational load than ω_f. The computation of whether the SC is interior to the target body ($r^{\mathtt{A}} \in \mathscr{C}_{obs}$) or not can be done as below,

$$\text{Impact} = \left(\frac{x^2}{x^2_{ell}} + \frac{y^2}{y^2_{ell}} + \frac{z^2}{z^2_{ell}} \leq 1\right), \quad \mathscr{C}_{obs} = \mathscr{C}_{obs}(x_{ell}, y_{ell}, z_{ell}) \tag{A.14}$$

where, x, y, z are the components of $r^{\mathtt{A}}$ and x_{ell}, y_{ell}, z_{ell} are the major axes of the minimum volume ellipsoid.

The second safety constraint is the escape of the SC from the gravitational field of the target body. This calls for a simple implementation of the position of the SC being within the Hill's sphere radius of the target body and additionally trajectories with radii above a certain range also will be included in this constraint to prevent violating scientific objectives of observation.

$$\text{Escape} = \left(r^{\mathtt{A}} \cdot r^{\mathtt{A}} \leq r^2_{esc}\right) \tag{A.15}$$

where, r_{esc} is a predefined maximum allowable magnitude of the SC position for any trajectory.

Imaging Constraints
The main objective of the mapping phase is to observe the target body and generate an accurate shape of it, whilst building crater maps for the HDA system. It also should be able to select a number of safe landing sites for the descent phase with this observation. Hence the goal fulfilment of the mapping algorithm pertains to the time of imaging a point of interest and observational quality for that time. These constraints have been derived in Section 6.4. Converting them to upper and lower bounds to represent the measure function, we get,

$$\text{Imageable} = (a_{min} < a_k < a_{max}) \quad \forall a \in \{\iota, \delta, \omega, \rho\}_k \tag{A.16}$$

where, ι is the LOS angle, δ is the tilt angle, ω is the angular rate and ρ is the range of the SC. These constraints have to be fulfilled for a duration of Δt_{obs} in order to achieve the required quality of observation. The completion status of the observation needs to be converted to a measure function as well, to keep track of the progress overall tasks.

$$\dot{mc}_k(\boldsymbol{x}, t) = \begin{cases} 1/\Delta t_{obs,k} & \text{If Imageable}_\text{k} \text{ and } mc_k<1 \\ 0 & \text{otherwise} \end{cases} \tag{A.17}$$

such that $mc_k \in [0,1]$ and the complete progress of observation can be given by $mc = \sum_{k=1}^{K} mc_k$. Therefore, for $mc = K$, the mission will be considered to be complete.

A.5.2. Mapping Initialisation Plots

The initialisation of the sample based model predictive optimisation has been implemented. The first stage of initialisation is the generation of a sample space for trajectory initialisation, $\mathscr{C}_{free}$, with a sample space of control actions, $\mathscr{U}$. The trajectory propagation until impact is considered. Each sample from the designated $\mathscr{C}_{free}$ is propagated with each action in $\mathscr{U}$. Since, an illumination model has been implemented, the first set of selected samples from $\mathscr{C}_{free}$, can be biased. The following plots hold a simplified version of this initialisation, for the reader to visualise the process. A trajectory from each sample has been plotted, until its impact, using a random control action from $\mathscr{U}$. $\mathscr{C}_{free}$, is built using the polyhedron structure of the asteroid, raising it by an altitude of 50 km and reducing the resolution of the polyhedron structure by 10^{-4}. This number can be increased or decreased based on heuristics, which have not been implemented due to time constraints. A spherical gravity model has been used to propagate the trajectories. A pre-selected number of samples of 50 for both $\mathscr{C}_{free}$ and $\mathscr{U}$ is taken for better visualisation. $\mathscr{U} \in 18$ m/s is enforced. A minimum volume ellipse for Kleopatra based on Kachiyan algorithm, is modelled for the impact constraint. Figure A.5 shows these initialised samples in 3D, while Figure A.7 shows the 2D plots for the same. The colour bar represents least to maximum illumination. Figure A.6 is shown to prove the algorithm stops propagation on impact.

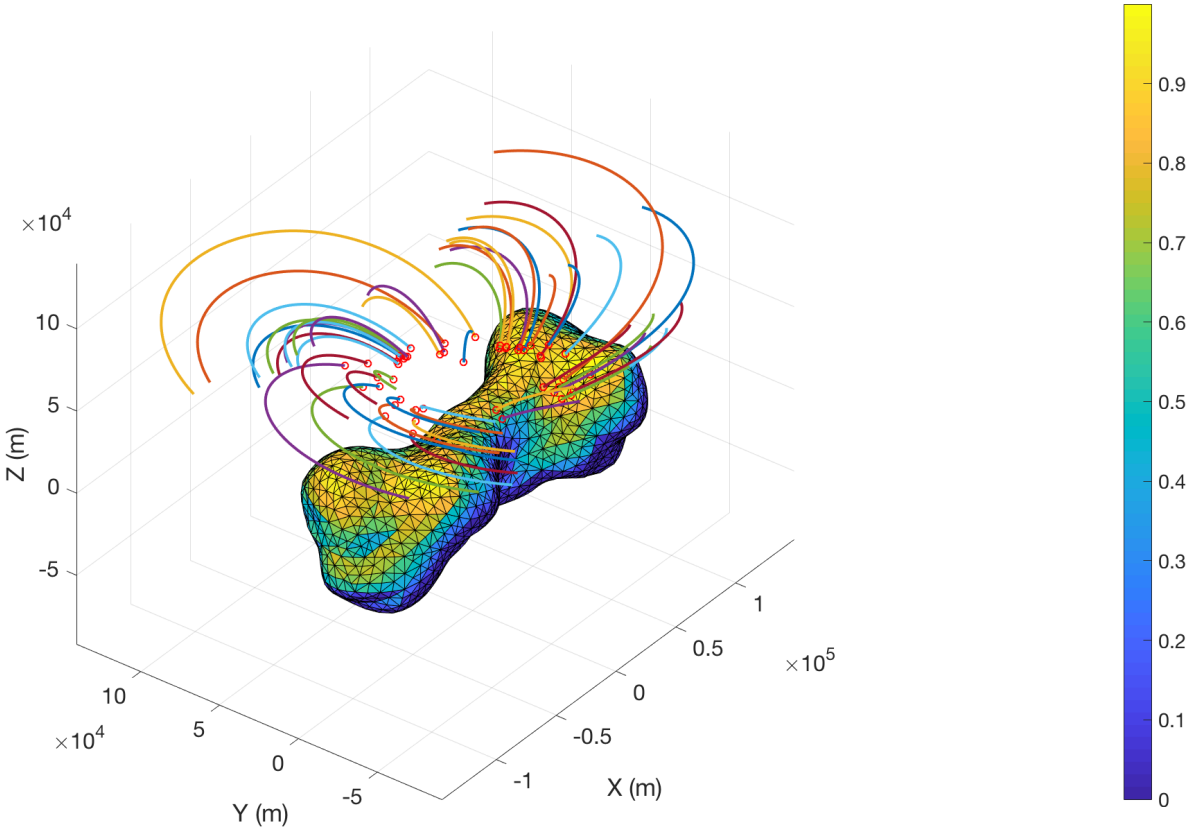


Figure A.5: Propagation of selected samples, with a randomly selected control action for each sample.

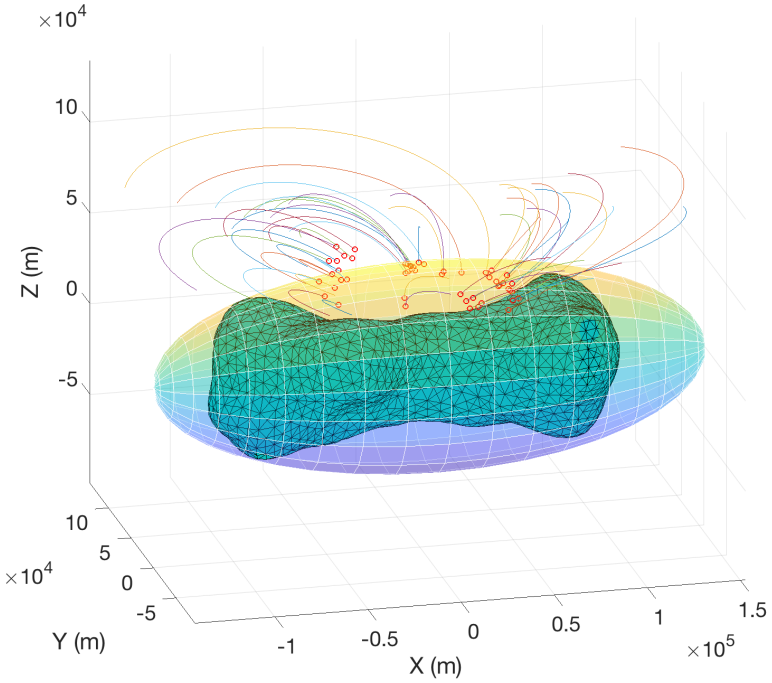


Figure A.6: Propagation of selected samples, with a minimum volume ellipsoid as impact criteria.

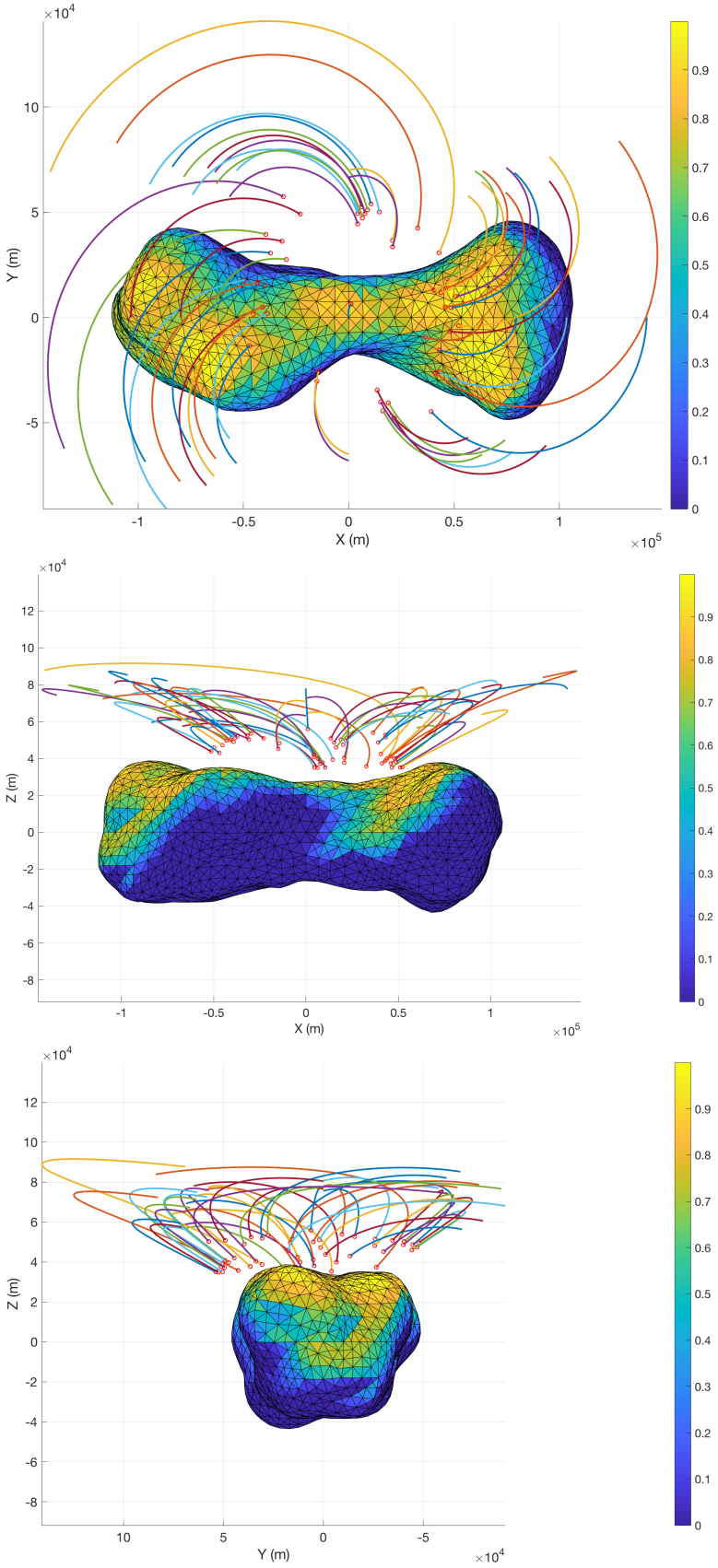


Figure A.7: 2D view of the initialised samples.

B

Derivations of the Dynamics for 6-DOF Mars Descent Test Case

Szmuk et al., 2017 have used successive convexification for the non-convex minimum-fuel OCP for 6 DOF rocket powered Mars landing Scenario. We use this case to verify our SCvx algorithm. It is a fixed final time problem and in the inertial and body reference frames. They use the Up-East-North frame as their inertial frame and hence, the rest of the this appendix, uses the same for easier understanding. The vehicle has a single engine and is capable of commanding various thrust magnitudes at different engine gimbal angles. The gravity, COM and inertia are assumed to remain constant. The state vector and its differential is given by,

$$\boldsymbol{x}(t) = \begin{pmatrix} m(t) \\ \boldsymbol{r}^{\mathtt{I}}(t) \\ \boldsymbol{v}^{\mathtt{I}}(t) \\ \boldsymbol{q}_{\mathtt{B/I}}(t) \\ \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \\ \boldsymbol{T}^{\mathtt{B}}(t) \\ \dot{\boldsymbol{T}}^{\mathtt{B}}(t) \end{pmatrix} \quad \dot{\boldsymbol{x}}(t) = \begin{pmatrix} -\alpha \| \boldsymbol{T}^{\mathtt{B}}(t) \| \\ \boldsymbol{v}^{\mathtt{I}}(t) \\ \boldsymbol{a}^{\mathtt{I}}(t) \\ \dot{\boldsymbol{q}}_{\mathtt{B/I}}(t) \\ \dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/I}}(t) \\ \dot{\boldsymbol{T}}^{\mathtt{B}}(t) \\ \boldsymbol{u}(t) \end{pmatrix} = \begin{pmatrix} -\alpha \| \boldsymbol{T}^{\mathtt{B}}(t) \| \\ \boldsymbol{v}^{\mathtt{I}}(t) \\ \frac{1}{m(t)} \boldsymbol{C}_{\mathtt{I/B}} \boldsymbol{T}^{\mathtt{B}}(t) + \boldsymbol{g}^{\mathtt{I}} \\ \frac{1}{2} \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \otimes \boldsymbol{q}_{\mathtt{B/I}}(t) \\ J^{-1}[\boldsymbol{r}^{\mathtt{B}}_{e} \times] \boldsymbol{T}^{\mathtt{B}}(t) - [\boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t) \times] J \boldsymbol{\omega}^{\mathtt{B}}_{\mathtt{B/I}}(t)) \\ \dot{\boldsymbol{T}}^{\mathtt{B}}(t) \\ \boldsymbol{u}(t) \end{pmatrix} \tag{B.1}$$

The non-linear dynamics are linearised by Taylor series to get the form as discussed in Eqs. (6.47) to (6.49). The linearised equations are given below,

$$\begin{aligned} f_{\boldsymbol{T}^{\mathtt{B}},t} &= \left(\| \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \| + \frac{\bar{\boldsymbol{T}}^{\mathtt{B}}(t)}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \|} (\boldsymbol{T}^{\mathtt{B}}(t) - \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \right) \\ f_{\boldsymbol{a}^{\mathtt{I}},t} &= \bar{\boldsymbol{a}}^{\mathtt{I}}(t) + \frac{\partial \boldsymbol{a}^{\mathtt{I}}}{\partial \boldsymbol{x}} \Big|_{\bar{\boldsymbol{x}}(t)} (\boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t)) \\ f_{\dot{\boldsymbol{q}}_{\mathtt{B/I}},t} &= \dot{\bar{\boldsymbol{q}}}_{\mathtt{B/I}}(t) + \frac{\partial \dot{\boldsymbol{q}}_{\mathtt{B/I}}}{\partial \boldsymbol{x}} \Big|_{\bar{\boldsymbol{x}}(t)} (\boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t)) \\ f_{\dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/I}},t} &= \dot{\bar{\boldsymbol{\omega}}}^{\mathtt{B}}_{\mathtt{B/I}}(t) + \frac{\partial \dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/I}}}{\partial \boldsymbol{x}} \Big|_{\bar{\boldsymbol{x}}(t)} (\boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t)) \end{aligned} \tag{B.2}$$

Now we will derive the ***linearisation matrices for continuous time***, for which we first derive the partial differentials with respect to state and controls.

Mass Dynamics:

$$\dot{m}(t) = -\alpha \left(\| \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \| + \frac{\bar{\boldsymbol{T}}^{\mathtt{B}}(t)}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \|} \left(\boldsymbol{T}^{\mathtt{B}}(t) - \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \right) \right) \tag{B.3}$$

The partial differential becomes,

$$\boldsymbol{\xi}_{\boldsymbol{T}} = -\alpha \frac{(\bar{\boldsymbol{T}}^{\mathtt{B}})'(t)}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \|} \tag{B.4}$$

Translational Acceleration:

$$\begin{aligned}\boldsymbol{a}^{\mathrm{I}}(t) &= \frac{1}{m(t)}\boldsymbol{C}_{\mathrm{I/B}}(t)\boldsymbol{T}^{\mathrm{B}}(t) + \boldsymbol{g}^{\mathrm{I}} \\ &= \frac{1}{m(t)}\Big[\boldsymbol{q}^{*}_{\mathrm{B/I}}(t) \otimes \boldsymbol{T}^{\mathrm{B}}(t) \otimes \Big]\boldsymbol{q}_{\mathrm{B/I}}(t) + \boldsymbol{g}^{\mathrm{I}} \\ &= \frac{1}{m(t)}\Big[\boldsymbol{T}^{\mathrm{B}}(t) \otimes \boldsymbol{q}_{\mathrm{B/I}}(t) \odot \Big]\boldsymbol{q}^{*}_{\mathrm{B/I}}(t) + \boldsymbol{g}^{\mathrm{I}}\end{aligned} \tag{B.5}$$

We find the partial differentials below,

$$\begin{aligned}\boldsymbol{\Psi}_m &= \frac{\partial \boldsymbol{a}^{\mathrm{I}}}{\partial m}\bigg|_{\bar{\boldsymbol{x}}} = \frac{1}{\bar{m}^2(t)}\bar{\boldsymbol{C}}_{\mathrm{I/B}}(t)\bar{\boldsymbol{T}}^{\mathrm{B}} \\ \boldsymbol{\Psi}_q &= \frac{\partial \boldsymbol{a}^{\mathrm{I}}}{\partial \boldsymbol{q}_{\mathrm{B/I}}}\bigg|_{\bar{\boldsymbol{x}}} = \frac{1}{\bar{m}(t)}\Big[\bar{\boldsymbol{q}}^{*}_{\mathrm{B/I}}(t) \otimes \bar{\boldsymbol{T}}^{\mathrm{B}}(t) \otimes \Big](1\ 1\ 1\ 1)^{\mathrm{T}} + \frac{1}{\bar{m}(t)}\Big[\bar{\boldsymbol{T}}^{\mathrm{B}}(t) \otimes \bar{\boldsymbol{q}}_{\mathrm{B/I}}(t) \odot \Big](-1\ -1\ -1\ 1)^{\mathrm{T}} \\ \boldsymbol{\Psi}_T &= \frac{\partial \boldsymbol{a}^{\mathrm{I}}}{\partial \boldsymbol{T}^{\mathrm{B}}}\bigg|_{\bar{\boldsymbol{x}}} = \frac{1}{\bar{m}(t)}\bar{\boldsymbol{C}}_{\mathrm{I/B}}(t)\end{aligned} \tag{B.6}$$

Attitude Kinematics:

$$\begin{aligned}\dot{\boldsymbol{q}}_{\mathrm{B/I}}(t) &= \frac{1}{2}\Big[\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}(t) \otimes \Big]\boldsymbol{q}_{\mathrm{B/I}}(t) \\ &= \frac{1}{2}\Big[\boldsymbol{q}_{\mathrm{B/I}}(t) \odot \Big]\begin{pmatrix}\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}(t) \\ 0\end{pmatrix}\end{aligned} \tag{B.7}$$

We find the partial differentials below,

$$\begin{aligned}\boldsymbol{\Omega}(\boldsymbol{\omega}) &= \frac{\partial \dot{\boldsymbol{q}}_{\mathrm{B/I}}}{\partial \boldsymbol{q}_{\mathrm{B/I}}}\bigg|_{\bar{\boldsymbol{x}}} = \frac{1}{2}\Big[\bar{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}}(t) \otimes \Big] \\ \boldsymbol{\Xi}(\boldsymbol{q}) &= \frac{\partial \dot{\boldsymbol{q}}_{\mathrm{B/I}}}{\partial \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}}\bigg|_{\bar{\boldsymbol{x}}} = \frac{1}{2}\Big[\bar{\boldsymbol{q}}_{\mathrm{B/I}}(t) \odot \Big]\end{aligned} \tag{B.8}$$

Attitude Dynamics:

$$\begin{aligned}\dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}}(t) &= J^{-1}\bigg([\boldsymbol{r}^{\mathrm{B}}_{e}\times]\boldsymbol{T}^{\mathrm{B}}(t) - \Big[\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}(t) \times \Big]J\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}(t)\bigg) \\ &= J^{-1}\bigg([\boldsymbol{r}^{\mathrm{B}}_{e}\times]\boldsymbol{T}^{\mathrm{B}}(t) + \Big[J\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}(t) \times \Big]\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}(t)\bigg)\end{aligned} \tag{B.9}$$

We find the partial differentials below,

$$\begin{aligned}\boldsymbol{\Theta}_{\omega} &= \frac{\partial \dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}}}{\partial \boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{B/I}}}\bigg|_{\bar{\boldsymbol{x}}} = J^{-1}\bigg(-\Big[\bar{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}}(t) \times \Big]J + \Big[J\bar{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}}(t)) \times \Big]\bigg) \\ \boldsymbol{\Theta}_{T} &= \frac{\partial \dot{\boldsymbol{\omega}}^{\mathrm{B}}_{\mathrm{B/I}}}{\partial \boldsymbol{T}^{\mathrm{B}}}\bigg|_{\bar{\boldsymbol{x}}} = J^{-1}[\boldsymbol{r}^{\mathrm{B}}_{e}\times]\end{aligned} \tag{B.10}$$

The compiled $\boldsymbol{A}(t)$ matrix is given below along with the control matrix. Since the state differential has controls as the last vector, it can be directly computed as given in $\boldsymbol{B}(t)$. And for $\boldsymbol{z}(t)$, the continuous time matrix is as follows,

$$\boldsymbol{A}(t) = \begin{bmatrix} 0 & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times4} & \boldsymbol{0}_{1\times3} & \boldsymbol{\xi}_{\boldsymbol{T}} & \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times4} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{\Psi}_m & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{\Psi}_{\boldsymbol{q}} & \boldsymbol{0}_{3\times3} & \boldsymbol{\Psi}_{\boldsymbol{T}} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{4\times1} & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{\Omega}(\boldsymbol{\omega}) & \boldsymbol{\Xi}(\boldsymbol{q}) & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} \\ \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times4} & \boldsymbol{\Theta}_{\boldsymbol{\omega}} & \boldsymbol{\Theta}_{\boldsymbol{T}} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times4} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times4} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}, \qquad \boldsymbol{B}(t) = \begin{bmatrix}\boldsymbol{0}_{17\times3} \\ \boldsymbol{I}_3\end{bmatrix}, \tag{B.11}$$

$$\boldsymbol{z}(t) = \begin{pmatrix} -\alpha \| \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \| \\ \bar{\boldsymbol{v}}^{\mathtt{I}}(t) \\ \frac{1}{\bar{m}(t)} \bar{\boldsymbol{C}}_{\mathtt{I}/\mathtt{B}} \bar{\boldsymbol{T}}^{\mathtt{B}}(t) + \boldsymbol{g}^{\mathtt{I}} \\ \frac{1}{2} \bar{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B}/\mathtt{I}}(t) \otimes \bar{\boldsymbol{q}}_{\mathtt{B}/\mathtt{I}}(t) \\ J^{-1}[\boldsymbol{r}_e^{\mathtt{B}} \times] \bar{\boldsymbol{T}}^{\mathtt{B}}(t) - [\bar{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B}/\mathtt{I}}(t) \times] J \bar{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B}/\mathtt{I}}(t)) \\ \dot{\bar{\boldsymbol{T}}}^{\mathtt{B}}(t) \\ \bar{\boldsymbol{u}}(t) \end{pmatrix} - \boldsymbol{A}(t) \begin{pmatrix} \bar{m}(t) \\ \bar{\boldsymbol{r}}^{\mathtt{I}}(t) \\ \bar{\boldsymbol{v}}^{\mathtt{I}}(t) \\ \bar{\boldsymbol{q}}_{\mathtt{B}/\mathtt{I}}(t) \\ \bar{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B}/\mathtt{I}}(t) \\ \bar{\boldsymbol{T}}^{\mathtt{B}}(t) \\ \dot{\bar{\boldsymbol{T}}}^{\mathtt{B}}(t) \end{pmatrix} - \boldsymbol{B}(t) \bar{\boldsymbol{u}}(t) \tag{B.12}$$

From TPBVP problem formulation, we have the initial and final required state values, which give us the $\boldsymbol{x}_0$ and $\boldsymbol{x}_f$ for a closed time, t_f problem. The in between states and state differential solutions can be found by discretisation. For discretisation the total time is divided into K nodes and the solutions at these points are calculated as per the Eqs. (6.83) and (6.84). The discretisation is done using the methodology explained in Chapter 6. The initialisation consists for the SCvx algorithm is done as below and compute the discrete matrices for the first iteration using the equations discussed in Chapter 6. For $k \in [0, K]$, first compute,

$$\begin{aligned} \bar{m}_k &= \left(\frac{K-k}{K}\right) m_{wet} + \left(\frac{k}{K}\right) m_{dry} \\ \bar{\boldsymbol{r}}^{\mathtt{I}}_k &= \left(\frac{k-k}{K}\right) \boldsymbol{r}^{\mathtt{I}}_0 + \left(\frac{k}{K}\right) \boldsymbol{r}^{\mathtt{I}}_f \\ \bar{\boldsymbol{v}}^{\mathtt{I}}_k &= \left(\frac{K-k}{K}\right) \boldsymbol{v}^{\mathtt{I}}_0 + \left(\frac{k}{K}\right) \boldsymbol{v}^{\mathtt{I}}_f \\ \bar{\boldsymbol{q}}_{\mathtt{B}/\mathtt{I}_k} &= [0\ 0\ 0\ 1]^{\mathtt{T}} \\ \bar{\boldsymbol{T}}^{\mathtt{B}}_k &= \bar{m}_k \boldsymbol{g} \boldsymbol{e}_z \mathtt{I} \\ \bar{\boldsymbol{x}}^0_k &= [\bar{m}_k\ \bar{\boldsymbol{r}}^{\mathtt{I}}_k\ \bar{\boldsymbol{v}}^{\mathtt{I}}_k\ \bar{\boldsymbol{q}}_{\mathtt{B}/\mathtt{I}_k}\ \boldsymbol{0}\ \bar{\boldsymbol{T}}^{\mathtt{B}}_k\ \boldsymbol{0}]^{\mathtt{T}}, \quad \bar{\boldsymbol{u}}^0_k = \boldsymbol{0} \end{aligned} \tag{B.13}$$

Then for $k \in [0, K)$, compute $\tilde{\boldsymbol{A}}^0_k$, $\tilde{\boldsymbol{B}}^0_k$, $\tilde{\boldsymbol{z}}^0_k$.

The ***constraints*** for this descent problem are stated below. The mass of the vehicle at any instance of time has to be greater than the dry mass of the vehicle,

$$-m_k \leq -m_{dry} \tag{B.14}$$

The angular rate of the vehicle at any instance of time has to be lesser than the maximum allowed angular rate,

$$\begin{aligned} \| \omega^{\mathtt{B}}_{\mathtt{B}/\mathtt{I}_k} \|_2 &\leq \omega_{max} \\ \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \omega^{\mathtt{B}}_{\mathtt{B}/\mathtt{I}_k} \right\|_2 &\leq \omega_{max} \end{aligned} \tag{B.15}$$

The thrust magnitude is bounded by upper and lower bounds,

$$T_{min} \leq \| \boldsymbol{T}^{\mathtt{B}}_k \|_2 \leq T_{max} \tag{B.16}$$

Splitting the constraint to a lower bound and upper bound we get,

$$\begin{aligned} T_{min} \leq \| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2 + \frac{\bar{\boldsymbol{T}}^{\mathtt{B}^{\mathtt{T}}}_k}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2} (\boldsymbol{T}^{\mathtt{B}}_k - \bar{\boldsymbol{T}}^{\mathtt{B}}_k) \leq T_{max} \\ -\left(\frac{\bar{\boldsymbol{T}}^{\mathtt{B}^{\mathtt{T}}}_k}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2} \boldsymbol{T}^{\mathtt{B}}_k \right) \leq -\left(T_{min} - \| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2 + \frac{\bar{\boldsymbol{T}}^{\mathtt{B}^{\mathtt{T}}}_k}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2} \bar{\boldsymbol{T}}^{\mathtt{B}}_k \right) \\ \left(\frac{\bar{\boldsymbol{T}}^{\mathtt{B}^{\mathtt{T}}}_k}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2} \boldsymbol{T}^{\mathtt{B}}_k \right) \leq \left(T_{max} - \| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2 + \frac{\bar{\boldsymbol{T}}^{\mathtt{B}^{\mathtt{T}}}_k}{\| \bar{\boldsymbol{T}}^{\mathtt{B}}_k \|_2} \bar{\boldsymbol{T}}^{\mathtt{B}}_k \right) \end{aligned} \tag{B.17}$$

The position vector of the vehicle at any point during the descent should stay above the glide-slope cone with a given minimum glide-slope angle, θ_{gs} above the horizon.

$$\begin{aligned} e^{\mathtt{I}}_z \boldsymbol{r}^{\mathtt{I}}_k &\geq \tan\theta_{gs} \left\| [e^{\mathtt{I}}_x \quad e^{\mathtt{I}}_y]^{\mathtt{T}} \boldsymbol{r}^{\mathtt{I}}_k \right\|_2 \\ \left\| \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{r}^{\mathtt{I}}_k \right\|_2 &\leq \begin{bmatrix} \frac{1}{\tan\theta_{gs}} & 0 & 0 \end{bmatrix} \boldsymbol{r}^{\mathtt{I}}_k \end{aligned} \tag{B.18}$$

The tilt angle, θ_t of the vehicle, i.e., the angle between the B frame and I frame has to be within an allowed maximum angle.

$$\begin{aligned} \cos\theta_{t_{max}} &\leq 1 - 2[\boldsymbol{q}_{2_k}^2 + \boldsymbol{q}_{3_k}^2] \\ \left\| \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{q}_{\mathtt{B}/\mathtt{I}_k} \right\|_2 &\leq \sqrt{\frac{1-\cos\theta_{t_{max}}}{2}} \end{aligned} \tag{B.19}$$

The engine gimbal angle, θ_g has to be constrained by a maximum angle.

$$\begin{aligned} \cos\theta_{g_{max}} \| \boldsymbol{T}_k^{\mathtt{B}} \|_2 &\leq e_z^{\mathtt{I}} \boldsymbol{T}_k^{\mathtt{B}} \\ \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{T}_k^{\mathtt{B}} \right\|_2 &\leq \begin{bmatrix} 0 & 0 & \frac{1}{\cos\theta_{g_{max}}} \end{bmatrix} \boldsymbol{T}_k^{\mathtt{B}} \end{aligned} \tag{B.20}$$

For the trust region, the constraint is the same as discussed earlier,

$$\delta \boldsymbol{x}_k^{\mathtt{T}} \delta \boldsymbol{x}_k + \delta \boldsymbol{u}_k^{\mathtt{T}} \delta \boldsymbol{u}_k \leq \eta_k \tag{B.21}$$

Let, $\boldsymbol{X}_k$ be the stacked state vector with states and controls, then the trust region constraint becomes as discussed earlier

$$\begin{aligned} \left(\boldsymbol{X}_k - \bar{\boldsymbol{X}}_k\right)^{\mathtt{T}} \left(\boldsymbol{X}_k - \bar{\boldsymbol{X}}_k\right) &\leq \eta_k \\ \left\| \begin{matrix} \left(1 - 2\bar{\boldsymbol{X}}_k^{\mathtt{T}} \boldsymbol{X}_k + (\bar{\boldsymbol{X}}_k^{\mathtt{T}} \bar{\boldsymbol{X}}_k - \eta_k)\right)/2 \\ \boldsymbol{X}_k \end{matrix} \right\|_2 &\leq \left(1 + 2\bar{\boldsymbol{X}}_k^{\mathtt{T}} \boldsymbol{X}_k - (\bar{\boldsymbol{X}}_k^{\mathtt{T}} \bar{\boldsymbol{X}}_k - \eta_k)\right)/2 \end{aligned} \tag{B.22}$$

And the ultimate constraints on the virtual control,

$$\left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\nu}_k \right\|_2 \leq s_k \tag{B.23}$$

We initialise η_k and ζ_k with vectors $\in \mathbb{R}^K$ and $\in \mathbb{R}^{K-1}$ respectively. The ***cost function*** for this problem is,

$$\min \; -m_K + w_\nu \| s_k \|_2 + w_\eta \| \eta_k \|_2 \tag{B.24}$$

The verication of this test case is presented in Section 7.5.2.

C

Mathematical Properties

This appendix, holds some basic mathematical properties of quaternions. It also holds comparisons between the usage of Euclidean or first norms in cost functions. A brief description of attitude representation methods using DCMs, Euler angles and Rodrigues parameters has also been discussed. Lastly, a tradeoff for available MATLAB® integrators has been put forth. This appendix is for the reader to get a brief idea of these concepts and the references should be looked into for in depth understanding.

C.1. Quaternion Properties

$\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are quaternions and $\gamma \in \mathbb{R}$

$$\begin{aligned} &1.\ \boldsymbol{a}\otimes(\boldsymbol{b}+\boldsymbol{c}) = \boldsymbol{a}\otimes\boldsymbol{b}+\boldsymbol{a}\otimes\boldsymbol{c} && 2.\ (\boldsymbol{a}\otimes\boldsymbol{b})^* = \boldsymbol{b}^*\otimes\boldsymbol{a}^* \\ &3.\ (\gamma\boldsymbol{a})\otimes\boldsymbol{b} = \boldsymbol{a}\otimes\gamma\boldsymbol{b}+\gamma(\boldsymbol{a}\otimes\boldsymbol{b}) && 4.\ \boldsymbol{a}\otimes(\boldsymbol{b}\otimes\boldsymbol{c}) = (\boldsymbol{a}\otimes\boldsymbol{b})\otimes\boldsymbol{c} \\ &5.\ \boldsymbol{a}^T(\boldsymbol{b}\otimes\boldsymbol{c}) = \boldsymbol{c}^T(\boldsymbol{b}^*\otimes\boldsymbol{a}) = \boldsymbol{b}^T(\boldsymbol{a}\otimes\boldsymbol{c}^*) \end{aligned} \tag{C.1}$$

Unit Quaternion Triple Identity

$$(\boldsymbol{t}\otimes\boldsymbol{q})^T(\boldsymbol{y}\otimes\boldsymbol{q}) = (\boldsymbol{q}\otimes\boldsymbol{t})^T(\boldsymbol{q}\otimes\boldsymbol{y}) = \boldsymbol{t}^T\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{t} \tag{C.2}$$

C.2. Attitude Representation Methods

C.2.1. Direction Cosine Matrices (DCMs) and Euler Angles

A direction cosine matrix or rotation matrix or coordinate transformation matrix as the name suggests is a matrix whose multiplication with a vector rotates the vector whilst preserving its length. The elements of a DCM are the cosines of the unsigned angles between the reference frames, that need to be rotated to each other. Consider the asteroid reference frame (A) and the body reference frame (B) as shown in Figure C.1. They can be represented respectively in the vector form by their coordinate axes, as,

$$\boldsymbol{a} = \hat{a}_1 + \hat{a}_2 + \hat{a}_3 \tag{C.3}$$

$$\boldsymbol{b} = \hat{b}_1 + \hat{b}_2 + \hat{b}_3 \tag{C.4}$$

The body reference frame in terms of the rotating reference frame can be expressed, as follows,

$$\hat{b}_1 = C_{11}\hat{a}_1 + C_{12}\hat{a}_2 + C_{13}\hat{a}_3 \tag{C.5}$$

$$\hat{b}_2 = C_{21}\hat{a}_1 + C_{22}\hat{a}_2 + C_{23}\hat{a}_3 \tag{C.6}$$

$$\hat{b}_3 = C_{31}\hat{a}_1 + C_{32}\hat{a}_2 + C_{33}\hat{a}_3 \tag{C.7}$$

As can be seen from Figure the constants C_{ij} are the cosines of the angles between the axes of frame A with each axis of frame B. Therefore, the reference frame B can be given in terms of reference frame A, as,

$$\begin{pmatrix}\hat{b}_1\\ \hat{b}_2\\ \hat{b}_3\end{pmatrix} = \begin{bmatrix}C_{11} & C_{12} & C_{13}\\ C_{21} & C_{22} & C_{23}\\ C_{31} & C_{32} & C_{33}\end{bmatrix}\begin{pmatrix}\hat{a}_1\\ \hat{a}_2\\ \hat{a}_3\end{pmatrix} = \begin{bmatrix}\cos\theta_{11} & \cos\theta_{12} & \cos\theta_{13}\\ \cos\theta_{21} & \cos\theta_{22} & \cos\theta_{23}\\ \cos\theta_{31} & \cos\theta_{32} & \cos\theta_{33}\end{bmatrix}\begin{pmatrix}\hat{a}_1\\ \hat{a}_2\\ \hat{a}_3\end{pmatrix} = \boldsymbol{C}_{\mathrm{B/A}}\begin{pmatrix}\hat{a}_1\\ \hat{a}_2\\ \hat{a}_3\end{pmatrix} \tag{C.8}$$

The matrix $\boldsymbol{C}_{\mathtt{B/A}}$ is called the direction cosine matrix and it orthonormal in nature and hence has the following properties,

$$\begin{aligned} \det \boldsymbol{C}_{\mathtt{B/A}} &= 1 \\ \boldsymbol{C}_{\mathtt{B/A}}^{-1} &= \boldsymbol{C}_{\mathtt{A/B}} = \boldsymbol{C}_{\mathtt{B/A}}^{T} \\ \boldsymbol{C}_{\mathtt{B/A}} \boldsymbol{C}_{\mathtt{B/A}}^{T} &= I_3 \end{aligned} \tag{C.9}$$

The disadvantage of using a DCM directly, is that nine parameters need to be known, where only three are required for rotating the reference frame. Therefore, we require six constraints for using a DCM. Euler angle rotation or sequential rotations are used in order to get the DCM without requiring constraints.

To achieve the reference frame B from the frame A, we implement two intermediate frames $\mathtt{A}'$ and $\mathtt{A}''$ with orthogonal unit vectors $(\hat{a}_1' \ \hat{a}_2' \ \hat{a}_3')$ and $(\hat{a}_1'' \ \hat{a}_2'' \ \hat{a}_3'')$ respectively. The sequence is represented as, $\mathtt{A} \rightarrow \mathtt{A}'$, $\mathtt{A}' \rightarrow \mathtt{A}''$ and finally $\mathtt{A}'' \rightarrow \mathtt{B}$ where the first rotation is about the axis $\hat{a}_1$ by an angle of θ_3, then about $\hat{a}_2'$ by an angle of θ_2 and the end about $\hat{a}_3''$ by θ_1. The angles θ_1, θ_2 and θ_3 are called the Euler angles and the DCMs for these rotations can be given below,

$$\boldsymbol{C}_{\mathtt{A}'/\mathtt{A}} = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.10}$$

$$\boldsymbol{C}_{\mathtt{A}''/\mathtt{A}'} = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \tag{C.11}$$

$$\boldsymbol{C}_{\mathtt{B}/\mathtt{A}''} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \tag{C.12}$$

Combining these sequential rotations gives us the DCM for direct rotation from frame A to frame B by multiplying the three earlier DCMs given by Eqs. 4.8, 4.9 and 4.10,

$$\begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{pmatrix} = \boldsymbol{C}_{\mathtt{B}/\mathtt{A}''} \begin{pmatrix} \hat{a}_1'' \\ \hat{a}_2'' \\ \hat{a}_3'' \end{pmatrix} = \boldsymbol{C}_{\mathtt{B}/\mathtt{A}''} \, \boldsymbol{C}_{\mathtt{A}''/\mathtt{A}'} \begin{pmatrix} \hat{a}_1' \\ \hat{a}_2' \\ \hat{a}_3' \end{pmatrix} = \boldsymbol{C}_{\mathtt{B}/\mathtt{A}''} \, \boldsymbol{C}_{\mathtt{A}''/\mathtt{A}'} \, \boldsymbol{C}_{\mathtt{A}'/\mathtt{A}} \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{pmatrix} \tag{C.13}$$

Therefore, $\boldsymbol{C}_{\mathtt{B}/\mathtt{A}} = \boldsymbol{C}_{\mathtt{B}/\mathtt{A}''} \, \boldsymbol{C}_{\mathtt{A}''/\mathtt{A}'} \, \boldsymbol{C}_{\mathtt{A}'/\mathtt{A}}$. Sequential rotations are not commutative, since matrix multiplication is not commutative. Hence they need to be followed in a specific defined order. Considering the three axes of a frame, there are 12 possible sequences to achieve a particular orientation. Considering X, Y & Z as the axes, the sequences are: XYZ, XZY, YZX, YXZ, ZYX, ZXY and XYX, XZX, YXY, YZY, ZXZ, ZYZ. The first six are called Tait-Bryan angles and nex six are called proper Euler angles.

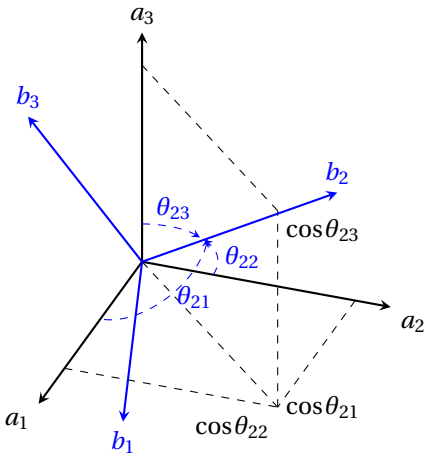


Figure C.1: Directions cosines visualised

DCMs using Euler angle rotations are the simplest method of incorporating a rotation between coordinate systems or reference frames and it can also be used for translation, scaling and projection as well. One major disadvantage of this method is the singularities at $\theta_2 = 90°/270°$ and $\theta_2 = 0°/180°$ for the first and second set of sequences, respectively, at the second axis of rotation. It is known as the gimbal lock and is caused due to infinite angular derivatives at those angles. At these angles, one of the two rotated axes replaces the other as it was in its previous orientation. Therefore, it is difficult to understand if the SC has achieved the desired new orientation after the rotation.

C.2.2. Rodrigues Parameters

Rodrigues parameters came as a solution to the constraint over the range of angles for the Euler-eigenaxis rotation. The initial solution was called the Classical Rodrigues Parameters (CRP). The constraint applied to the fourth element of the quaternion (Schaub and Junkins, 2009):

$$q_4 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2} = \cos\frac{\theta}{2} \tag{C.14}$$

With this the CRPs are defined as below:

$$\begin{pmatrix}\tau_1\\ \tau_2\\ \tau_3\end{pmatrix} = \begin{pmatrix}q_1/q_4\\ q_2/q_4\\ q_3/q_4\end{pmatrix} = \begin{pmatrix}e_1 \tan\theta/2\\ e_2 \tan\theta/2\\ e_3 \tan\theta/2\end{pmatrix} \tag{C.15}$$

As can be seen, that the above equations contain the holonomic constraint, but introduce a singularity at $\theta = \pm 180°$. To remove this the denominator of the CRPs are added by 1 and the singularity moves to $\theta = \pm 360°$, which signifies a full rotation. These new parameters are called Modified Rodrigues Parameters (MRPs).

$$\begin{pmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\end{pmatrix} = \begin{pmatrix}q_1/(1+q_4)\\ q_2/(1+q_4)\\ q_3/(1+q_4)\end{pmatrix} = \begin{pmatrix}e_1 \tan\theta/4\\ e_2 \tan\theta/4\\ e_3 \tan\theta/4\end{pmatrix} \tag{C.16}$$

It is possible to modify the MRPs, so that the singularity is shifted even more ahead to something like $\theta = \pm 720°$, but they reintroduce singularities in the kinematic equation at $\theta = \pm 360°$ and also increase the computational load, since the kinematics is defined by higher order polynomials and is no longer a simple set off second-order polynomials.

C.3. Ridge and Lasso Regression Comparison

The penalised term in ridge regression cost is an L2 norm, and in lasso regression cost is an L1 norm[1]. The fundamental difference between the two is, of course, that of proportionality of the prior to squared values and of the latter to absolute values and this gives them their benefits and drawbacks. Below is the list of compared properties of the two.

Robustness
Robustness in terms of the cost function would be its indifference to outliers, such that extreme values in the data are ignored. Since L1 norm considers these outliers linearly as compared to L2, which squares them, the prior makes a more robust cost function.

Stability
Stability is the indifference to a horizontal deviation of an outlier. Since L1 provides a linear solution, it is affected more by a horizontal deviation as compared to L2.

Solution Numeracy
The L2 norm dictates a unique solution, whereas the L1 norm would dictate a number of solutions having the same length but not in the same direction.

Sparsity
A vector/matrix is called sparse if it has few non-zero values. L1 norm has the inherent property of creating more coefficients with zeros or very small values as compared to L2 norms.

[1] https://www.kaggle.com/residentmario/l1-norms-versus-l2-norms, date accessed:02-10-18

Computational Difficulty
An analytical solution exists for the L2 norm but not L1 and hence makes the latter computationally more difficult. However, in the case of a sparse algorithm due to its inherent property of creating sparseness, the L1 norm proves to be more computationally efficient.

Amongst these properties, a compromise is made on the robustness and sparsity for stability and solution numeracy. Since the computational efficiency of the DQ based algorithm compared to the CQ algorithm is an objective we do not emphasise on the computational efficiency provided by L1 norms for the sparse problem at hand.

C.4. MATLAB® Integrators

Table C.1: Trade off of MATLAB® Integrators.[2]

Integrator	Problem Type	Accuracy
ode45	Nonstiff	Medium
ode23	Nonstiff	Low
ode113	Nonstiff	Low to high
ode15s	Stiff	Low to medium
ode23s	Stiff	Low
ode23t	Moderately Stiff	Low
ode23tb	Stiff	Low

Stiffness of a differential equation pertains to its incapability of choosing a an appropriate step size for finding a solution at a certain region. What this means is in regions of high variability the step size should be smaller and in regions of low variability the step size should be larger, which makes the equation nonstiff. Hence, selection of a nonstiff integrator, when unsure of the problem behaviour is a must. As can be seen from the table, ode45 is the best integrator to approach the problem first and in case of high inaccuracies from the exact solution, one should opt for ode1123.

[2]https://web.mit.edu/voigtlab/BP205/Notes/BP205_Matlab_handout.pdf, date accessed:07-03-19

D

Plots

D.1. Derived Linearised Dynamics Verification

These plots are for the test case presented in Section 7.5.1 for verification of the linearised dynamics. They show no variation in their components, since the angular rate, $\dot{\boldsymbol{\omega}}^{\mathtt{B}}_{\mathtt{B/I}} = \mathbf{0}_{3\times1}$.

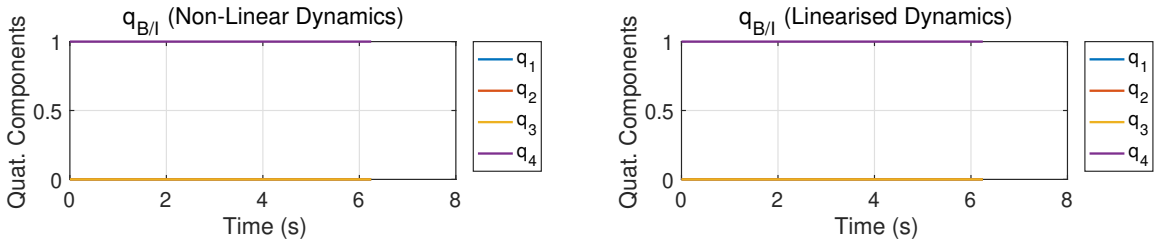


Figure D.1: Simulated quaternions in the I frame.

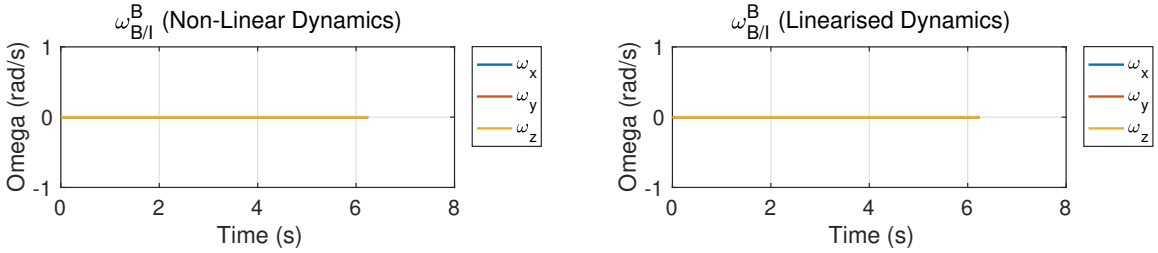


Figure D.2: Simulated angular velocities in the B frame.

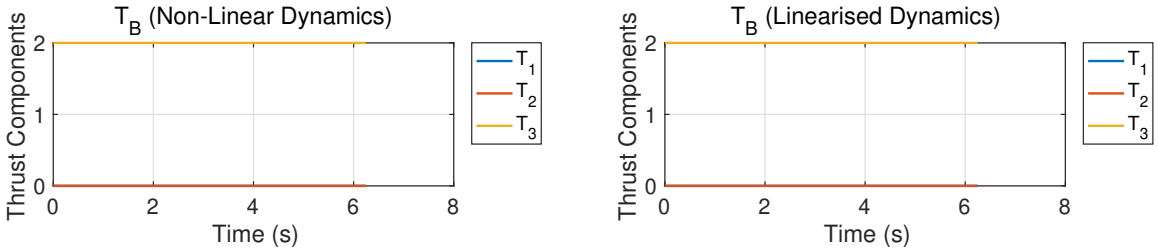


Figure D.3: Simulated torques in the B frame.

Bibliography

[1] Açıkmeşe, B., Carson, J., and Blackmore, L. "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem". *IEEE Transactions on Control Systems Technology* Vol. 21.No. 6 (2013), pp. 2104–2113. DOI: 10.1109/TCST.2012.2237346.

[2] Açıkmeşe, B. and Ploen, S. R. "Convex programming approach to powered descent guidance for Mars landing". *journal of Guidance, Control and Dynamics* Vol. 30.No. 5 (2007), pp. 1353–1366. DOI: 10.1109/TCST.2012.2237346.

[3] Antsaklis, P. and Michel, A. *Linear Systems*. Birkhäuser, Boston, 2006.

[4] Ballester, M. "Asteroid Mission Guidance and Control using Dual Quaternions". MSc. thesis. Technical University of Delft, 2018.

[5] Berry, K., Sutter, B., May, A., Williams, K., Barbee, B. W., Beckman, M., and Williams, B. "Osiris-rex touch-and-go (tag) mission design and analysis". *Advances in the Astronautical Sciences* Vol. 149 (2013), pp. 667–678.

[6] Beshore, E., Lauretta, D., Boynton, W., Shinohara, C., Sutter, B., Everett, D., GalEdd, J., Mink, R., Moreau, M., and Dworkin, J. "The OSIRIS-REx Asteroid Sample Return Mission". *IEEE* (2015).

[7] Blackmore, L. and Açıkmeşe B. Carson, J. and. "Lossless convexification of control constraints for a class of nonlinear optimal control problems". *American Control Conference* (2012).

[8] Boyce, E. and DiPrima, R. *Elementary Differential Equations and Boundary Value Problems*. John Wiley and Sons, Inc., 2008.

[9] Boyd, S. and Vandenberghe, L. *Convex Optimization*. Vol. 25. Cambridge University Press, 2010. DOI: 10.1080/10556781003625177.

[10] Canale, A. "Autonomous Spacecraft Guidance for Small-Body Proximity Missions". MSc. thesis. Technical University of Delft, 2018.

[11] Capolupo, F., Simeon, T., and Berges, J. "Heuristic Guidance Techniques for the Exploration of Small Celestial Bodies". *IFAC-PapersOnLine* Vol. 50.No. 1 (2017), pp. 8279–8284. DOI: 10.1016/j.ifacol.2017.08.1401.

[12] Daniilidis, K. "Hand-Eye Calibration Using Dual Quaternions". *Int. J. Robot. Res.* Vol. 18.No. 3 (1999), pp. 286–298.

[13] Dickinson, C., Daly, M., Barnouin, O., Bierhaus, B., Gaudreau, D., Tripp J.; Ilnicki, M., and Hildebrand, A. "An Overview of the OSIRIS REx Laser Altimeter (OLA)". *43rd Lunar and Planetary Science Conference* (2012).

[14] Domahidi, A., Chu, E., and Boyd, S. "ECOS: An SOCP solver for embedded systems". *European Control Conference (ECC)* (2013), pp. 3071–3076.

[15] Dong, H., Hu, Q., Friswell, M. I., and Ma, G. "Dual-Quaternion-Based Fault-Tolerant Control for Spacecraft Tracking With Finite-Time Convergence". *IEEE Transactions on Control Systems Technology* Vol. 25.No. 4 (2017), pp. 1231–1242. DOI: 10.1109/TCST.2016.2603070.

[16] Ecaterina, C., Matthew, B. W., Sai, P. K., Andrew, R. M., and Matthias, G. K. *A Comparative Evaluation of Matlab, Octave, FreeMat, Scilab, R, and IDL on Tara*. Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2012. DOI: 10.1.1.174.7589.

[17] ESA-ESTEC. "ECSS-Space Engineering-Satellite AOCS Requirements-60C". *ESA Requirements and Standards Division* (2013).

[18] Filipe, N. and Tsiotras, P. "Adaptive Model-Independent Tracking of Rigid Body Position and Attitude Motion with Mass and Inertia Matrix Identification using Dual Quaternions". *AIAA Guidance, Navigation, and Control (GNC) Conference* (2013), pp. 1–15. DOI: 10.2514/6.2013-5173.

[19] Fujiwara, A. et al. “The Rubble-Pile Asteroid Itokawa as Observed by Hayabusa”. *Science* Vol. 312 (2006), pp. 1330–1334. DOI: 10.1126/science.1125841.

[20] GalEdd, J. and Cheuvront, A. “The OSIRIS-REx Asteroid Sample Return Mission Operations Design ”. *IEEE* (2015).

[21] Geissler, P., Bottke, W. F., Nolan, M., Greenberg, R., Petit, J. M., Durda, D. D., Asphaug, E., and Head, J. “Collisional and dynamical history of Ida”. *Icarus* Vol. 120.No. 1 (1996), pp. 106–118. DOI: 10.1006/icar.1996.0040.

[22] Gerth, I. “Convex Optimization for Constrained and Unified Landing Guidance”. MSc. thesis. Technical University of Delft, 2014.

[23] Ghallab, M., Nau, D., and Traverso, P. *Automated Planning: Theory and Practice*. Elsevier Inc., 2004.

[24] Godard, B., Budnik, F., Muñoz, P., Morley, T., and Janarthanan, V. “Orbit Determination of Rosetta around Comet 67P/Churyumov-Gerasimenko”. *International Symposium on Space Flight Dynamics* (2015).

[25] Grandsen, D. and Mooij, E. “Simple Adaptive Control of a Satellite with Large Flexible Appendages”. *International Astronautical Congress* (2018).

[26] Hampton, D. L., Baer, J. W., Huisjen, M. A., Varner, C. C., Delamere, A., Wellnitz, D. D., A’Hearn, M. F., and Klaasen, K. P. “An Overview of the Instrument Suite for the Deep Impact Mission”. *Space Science Reviews* Vol. 117.No. 1 (2005), pp. 43–93. DOI: 10.1007/s11214-005-3390-8.

[27] Hapke, B. “Bidirectional reflectance spectroscopy. 4. The extinction coefficient and the opposition effect”. *Icarus 67* (1986), pp. 264–280.

[28] Harris, A. W., Fahnestock, E. G., and Pravec, P. “On the shapes and spins of "rubble pile" asteroids”. *Icarus* Vol. 199.No. 2 (2009), pp. 310–318. DOI: 10.1016/j.icarus.2008.09.012.

[29] Holsapple, K. A. “Equilibrium configurations of solid cohesionless bodies”. *Icarus* Vol. 154.No. 2 (2001), pp. 432–448. DOI: 10.1006/icar.2001.6683.

[30] Jia, Y. B. *Dual quaternion*. Com S 477/577, Fall 2015, Iowa State Univ. Course Handouts, Ames, IA, 2013.

[31] Jia, Y. B. *Quaternions*. Com S 477/577, Fall 2018, Iowa State Univ. Course Handouts, Ames, IA, 2018.

[32] Kavan, L., Collins, S., Zara, J., and O’Sullivan, C. “Skinning with dual quaternions”. *Proceedings of the 2007 symposium on Interactive 3D graphics and games* (2007), pp. 39–46. DOI: 10.1145/1230100.1230107.

[33] Kenwright, B. “A Beginners Guide to Dual-Quaternions: What They Are, How They Work, and How to Use Them for 3D Character Hierarchies”. *The 20th International Conference on Computer Graphics, Visualization and Computer Vision* (2012), pp. 1–13. DOI: 10.1109/CVPR.2015.7298631.

[34] Kubota, T., Hashimoto, T.i, Kawaguchi, J., Uo, M., and Shirakawa, K. “Guidance and navigation of Hayabusa spacecraft for asteroid exploration and sample return mission”. *2006 SICE-ICASE International Joint Conference* (2006), pp. 2793–2796. DOI: 10.1109/SICE.2006.314761.

[35] Kurt, B. and Yosi, S. “Penalty Functions and Constrained Optimization”. *Rose-Hulman Institute of Technology* (2005).

[36] Kwon, J. W., Lee, D., and Bang, H. “Virtual Trajectory Augmented Landing Control Based on Dual Quaternion for Lunar Lander”. *journal of Guidance, Control, and Dynamics* Vol. 39.No. 9 (2016), pp. 2044–2057. DOI: 10.2514/1.G001459.

[37] LaValle, S. “Motion planning: Part I: The essentials”. *IEEE Robotics and Automation Magazine* Vol. 18.No. 1 (2011).

[38] LaValle, S. *Planning Algorithms*. Cambridge University Press., 2006.

[39] Lee, U. and Mesbahi, M. “Constrained Autonomous Precision Landing via Dual Quaternions and Model Predictive Control”. *journal of Guidance, Control, and Dynamics* Vol. 40.No. 2 (2017), pp. 292–308. DOI: 10.2514/1.G001879.

[40] Lee, U. and Mesbahi, M. “Optimal Powered Descent Guidance with 6-DoF Line of Sight Constraints via Unit Dual Quaternions”. *AIAA Guidance, Navigation, and Control Conference* (2015), pp. 1–21. DOI: 10.2514/6.2015-0319.

[41] Ley, W., Wittmann, K., and Hallmann, W. *HandBook of Space Technology*. John Wiley & Sons, Ltd., 2011.

[42] Lissauer, J. and Pater, I. de. *Fundamental Planetary Science*. Cambridge University Press, New York, USA, 2013.

[43] Lyness, J. and Moler, C. "Numerical Differentiation of Analytic Functions". *SIAM journal on Numerical Analysis* Vol. 4.No. 2 (1967).

[44] Mao, Y., Szmuk, M., and Acıkmeşe, B. "Successive convexification of non-convex optimal control problems and its convergence properties". *2016 IEEE 55th Conference on Decision and Control, CDC 2016* (2016), pp. 3636–3641. DOI: 10.1109/CDC.2016.7798816.

[45] Markley, L. F. and Crassidis, J. L. *Fundamentals of Spacecraft Attitude Determination and Control*. Springer, 2013. DOI: 10.1007/978-1-4939-0802-8.

[46] Martins, J., Sturdza, P., and Alonso, J. "The Complex-Step Derivative Approximation". *ACM Transactions on Mathematical Software* Vol. 29.No. 9 (2003), pp. 245–262. DOI: 10.1109/TCST.2012.2237346.

[47] Maruya, M., Sawai, S., Kubota, T., Hashimoto, T., and Ninomiya, K. "Estimation of Motion and Shape of Asteroid Based on Image Sequences". *14th Int. Symposium on Space Flight Dynamics* (2000).

[48] Mastrodemos, N., Kubitschek, D. G., and Synnott, S. P. "Autonomous Navigation for the Deep Impact Mission Encounter with Comet Tempel 1". *Space Science Reviews* Vol. 117.No. 1 (2005), pp. 95–121. DOI: 10.1007/s11214-005-3394-4.

[49] McDonald, J. "Teaching quaternions is not complex". *Computer Graphics Forum* Vol. 29.No. 8 (2010), pp. 2447–2455. DOI: 10.1111/j.1467-8659.2010.01756.

[50] Montebruck, O. and Gill, E. *Satellite Orbits: Models, Methods and Applications*. American Institute of Aeronautics and Astronautics, 2008.

[51] Mooij, E. "The motion of a vehicle in a planetary atmosphere". *Delft University of Technology, Faculty of Aerospace Engineering* (1994).

[52] Muñoz, P., Budnik, F., Godard, B., Morley, T., Companys, V., Herfort, U., and Casas, C. "Preparations and Strategy for Navigation During Rosetta Comet Phase". *International Symposium on Space Flight Dynamics* (2012), pp. 1–17.

[53] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. "Shape Distributions". *ACM Transactions on Graphics* Vol. 21.No. 4 (2002).

[54] Park, R. S., Werner, R. A., and Bhaskaran, S. "Estimating Small-Body Gravity Field from Shape Model and Navigation Data". *journal of Guidance, Control, and Dynamics* Vol. 33.No. 1 (2010), pp. 212–221. DOI: 10.2514/1.41585.

[55] Pavone, M., Acıkmeşe, B., Nesnas, I. A., and Starek, J. "Spacecraft Autonomy Challenges for Next Generation Space Missions". *Springer Lecture Notes in Control and Information Sciences* (2014), pp. 1–34. DOI: 10.1007/978-3-662-47694-9_1.

[56] Pfeiffer, C. "An Analysis of Guidance Modes". Tech. rep. NASA Electronics Research Center (1968).

[57] Pinson, R. and Lu, P. "Trajectory Design Employing Convex Optimization for Landing on Irregularly Shaped Asteroids". *AIAA/AAS Astrodynamics Specialist Conference* (2016).

[58] Razgus, B. "Relative Navigation in Asteroid Missions: Dual Quaternion Approach". MSc. thesis. Technical University of Delft, 2017.

[59] Razgus, B., Mooij, E., and Choukroun, D. "Relative Navigation in Asteroid Missions Using Dual Quaternion Filtering". *journal of Guidance, Control, and Dynamics* Vol. 40.No. 9 (2017), pp. 1–16. DOI: 10.2514/1.G002805.

[60] Ridder, K. "Convex Guidance, navigation and Control for Pin-Point Lunar Landing". MSc. thesis. Technical University of Delft, 2016.

[61] Sagliano, M. "Pseudospectral Convex Optimization for Powered Descent and Landing". *journal of Guidance, Control, and Dynamics* (2017), pp. 1–15. DOI: 10.2514/1.G002818.

[62] Sagliano, M., Theil S. Bergsma, M., D'Onofrio, V., Whittle, L., and Viavattene, G. "On the Radau pseudospectral method: theoretical and implementation advances". *CEAS Space journal* (2017). DOI: 10.1007/s12567-017-0165-5.

[63] Sawai, S., Kubota, T., Hashimoto, T., Kawaguchi, J., and Fujiwara, A. "Robotics Technology for Asteroid Sample Return Mission MUSES-C". *6th International Symposium on Artificial Intelligence and Robotics & Automation in Space* (2001).

[64] Schaub, H. and Junkins, J. L. *Analytical Mechanics of Space Systems*. American Institute of Aeronautics and Astronautics, 2009.

[65] Scheeres, D. J. "Orbit Mechanics About Asteroids and Comets". *journal of Guidance, Control, and Dynamics* Vol. 35.No. 3 (2012), pp. 987–997. DOI: 10.2514/1.57247.

[66] Seabrook, J., Daly, M., Barnouin, O., Nair, A., Espiritu, R., Gaskell, R., Johnson, C, Bierhaus, E., and Lauretta, D. "Shape Model Construction of Bennu using the OSIRIS-REX Laser Altimeter (OLA)." *48th Lunar and Planetary Science Conference* (2017).

[67] Sheppard, S. S., Jewitt, D., and Kleyna, J. "A Survey for "Normal" Irregular Satellites around Neptune: Limits to Completeness". *The Astronomical Journal* Vol. 132.No. 1 (2006), pp. 171–176. DOI: 10.1086/504799.

[68] Sidi, M. J. *Spacecraft Dynamics and Control*. Cambridge University Press, 1997.

[69] Squire, W. and Trapp, G. "Using Complex Variables to Estimate Derivatives of Real Functions". *Society for Industrial and Applied Mathematics* Vol. 40.No. 1 (1998).

[70] Surovik, D. "Autonomous Mission Design in Extreme Orbit Environments". PhD thesis. University of Colorado, 2016.

[71] Surovik, D. A. and Scheeres, D. J. "Autonomous Maneuver Planning at Small Bodies via Mission Objective Reachability Analysis". *AIAA/AAS Astrodynamics Specialist Conference* (2014), pp. 1–12. DOI: 10.2514/6.2014-4147.

[72] Surovik, D. A. and Scheeres, D. J. "Non-Keplerian Trajectory Planning via Heuristic-Guided Objective Reachability Analysis" (2017).

[73] Szmuk, M. and Açıkmeşe, B. "Successive Convexification for Mars 6-DoF Rocket Powered Landing with Free-Final-Time". *AIAA Guidance, Navigation, and Control Conference* (2018).

[74] Szmuk, M., Utku, E., and Açıkmeşe, B. "Successive Convexification for Mars 6-DoF Powered Descent Landing Guidance". *AIAA Guidance, Navigation, and Control Conference.* (2017).

[75] Takahashi, Y. and Scheeres, D. J. *Small body surface gravity fields via spherical harmonic expansions*. Vol. 119. 2. 2014, pp. 169–206. DOI: 10.1007/s10569-014-9552-9.

[76] Tsuda, Y., Yoshikawa, M., Abe, M., Minamino, H., and Nakazawa, S. "System design of the hayabusa 2-asteroid sample return mission to 1999 JU3". *Acta Astronautica* Vol. 91 (2013), pp. 356–362. DOI: 10.1016/j.actaastro.2013.06.028.

[77] Tsuda, Y., Yoshikawa, M., Saiki, T., Nakazawa, S., and Watanabe, S. "Hayabusa2-Sample return and kinetic impact mission to near-earth asteroid Ryugu". *Acta Astronica* (2017).

[78] Utashima, M. "Spacecraft Orbits Around Asteroids for Global Mapping". *journal of Spacecraft and Rockets* Vol. 34.No. 2 (1997).

[79] Wakker, K. F. *Fundamentals of Astrodynamics*. Technical University of Delft, 2015.

[80] Wang, J. and Cui, N. "A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance". *AIAA Guidance, Navigation and Control* (2018).

[81] Wang, Z. and Grant, M. "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming". *journal of Guidance, Control and Dynamics* (2017).

[82] Werner, R. A. and Scheeres, D. J. "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia". *Celestial Mechanics and Dynamical Astronomy* Vol. 65.No. 3 (1997), pp. 313–344.

[83] Wie, B. *Space Vehicle Dynamics and Control*. Springer Verlag, Heidelberg, 2001.

[84] Woicke, S., Gonzalez, M., ElHajj, I., Mes, J., Henkel, M., Autar, R., and Klavers, R. "Comparison of Crater-Detection Algorithms for Terrain-Relative Navigation". *AIAA Guidance, Navigation, and Control Conference* (2018).

[85] Xinfu, L. and Ping, L. “Solving Nonconvex Optimal Control Problems by Convex Optimization”. *journal of Guidance, Control, and Dynamics* Vol. 37.No. 3 (2014).

[86] Xinfu, L., Ping, L., and Binfeng, P. “Survey of convex optimization for aerospace applications”. *Astrodynamics*. Vol. 1.No. 1 (2017).

[87] Yoshihide, S., Jozef, C., and Rievers, B. “Thermal Radiation Model for the Rosetta Spacecraft”. *AIAA Guidance, Navigation and Control Conference* (2010).

[88] Zhang, F. and Duan, G. “Robust Integrated Translation and Rotation Finite-Time Maneuver of a Rigid Spacecraft Based on Dual Quaternion”. *AIAA Guidance, Navigation, and Control Conference* No. 92 (2011), pp. 1–17. DOI: 10.2514/6.2011-6396.